Design Requirements to Integrate Eye Trackers in Simulation Environments: Aeronautical Use Case

Jean-Paul Imbert, Christophe Hurter, Vsevolod Peysakhovich, Colin Blättler, Frédéric Dehais and Cyril Camachon

Abstract Eye tracking (ET) provides various data like gaze position, pupil size, and eye movement events (blinks, fixations, saccades, etc.). These data can reveal users' cognitive/attentional state but also provide a worthwhile input to human-computer interfaces. Recording and processing such data is an issue especially when integrating ET systems within existing environments. The synchronization of events from the simulation environment and physiological measurement devices with ET data is also a concern. In this paper, we reflect upon a seamless integration process. We gather task fulfillment requirements and confront them with technical constraints. Based on this structured task analysis, we present architecture guidelines regarding efficient ET system integration. Finally, we provide a relevant use case of experimental environment where ET systems have been successfully integrated and discuss architecture solutions.

Keywords Eye tracking · Eye gaze analysis · Interaction · Human factors

J.-P. Imbert (✉) · C. Hurter
ENAC, Toulouse, France
e-mail: jean-paul.imbert@enac.fr

C. Hurter
e-mail: christophe.hurter@enac.fr

V. Peysakhovich · F. Dehais
ISAE, Toulouse, France
e-mail: vsevolod.peysakhovich@isae.fr

F. Dehais
e-mail: frederic.dehais@isae.fr

C. Blättler · C. Camachon
CReA, Salon-de-Provence, France
e-mail: colin.blattler@defense.gouv.fr

C. Camachon
e-mail: cyril.camachon@defense.gouv.fr

1   Introduction

The analysis of user visual scanpaths gives insights about the way a Human-Machine Interface (HMI) is used. Eye tracking (ET) allows for the collection of information to infer user activity and his/her cognitive workload [10]. Nowadays, eye trackers are also used as an input device, providing system a new way to point to or interact with an [5]. Though the advantages of ET systems are known and unanimously accepted, ET integration within existing systems remains a challenging task. Firstly, ET devices are mostly off-the-shelf products and need to be integrated in existing systems by the customers themselves. Such integration can be a problem especially when existing environment (i.e., flight or drive simulators) does not allow communication with third party software. Secondly, eye trackers produce large amount of data which need to be stored and then processed. When an ET is used as a system input the data must be processed in real- or near real-time, thus adding complications. To the best of our knowledge, no literature has previously tried to reflect upon such ET integration. In this paper we gathered task fulfillment needs and countered them with technical constraints. From this structured task analysis, we extracted design requirements. The task analysis was completed based on user interviews and our own experience with ET systems in aeronautics. The presented taxonomy and design guidelines would help practitioners to better understand the challenges and the technical solution to the integration of ET with simulation systems. Our contributions are a taxonomy of tasks, technical challenges, and design architecture requirements for ET integration. The remainder of the paper is structured as follows: first we present a review of existing work using ET systems; then we detail our taxonomy with tasks and technical challenges; and lastly we present a use case where we fulfilled our identified design requirements. For each example, we detail and explain the technical challenges. Finally we summarize our architecture recommendations and conclude with future challenges for ET systems integration.

2   Experimental Process and Eye Tracking

In a multi-factorial approach, many data can be collected from different sources: various psychophysiological sensors (eye tracker, electroencephalography, electrocardiography, functional near-infrared spectroscopy, etc.) and the experimental environment (HMI events such as mouse/keyboard input, simulator events, interaction with other participants). The data collected from these sources have to be synchronized for further analysis, for example to verify if an event in the experimentation scenario is associated with a fixation over a moving object of the interface. Data merging from different sources can be complex and time consuming depending on the architecture of the experimentation. For example, centralized time synchronization is mandatory while receiving data from multiple computers.

Another issue, when conducting human factors research, is related to physical integration of ET device within the experimental set up. Integration of such a system is highly dependent of the experimental context/constraint. Indeed, in applied cognitive research, the Cognitive System Engineering (CSE) framework [20] proposes four stages which achieve a different balance between ecological validity and experimental control: Stage (1) cognitive processes testing (initial laboratory experiments); (2) functions testing (laboratory methods within a basic context); (3) functions testing within complex simulations; (4) and behavioral observation within an experimental operational setting. These stages describe different architectures in terms of complexity. The first stage is appropriate when new concepts are studied. Usually a new application must be developed in order to test these concepts. Therefore, the research engineer in charge will be able to integrate the sensors in the developed ad-hoc architecture. An example of the second station would be serious games or microworlds [12]. When initial concepts are validated, a microworld environment can help test more complex functions in a simplified setup compared to complex simulation. As with the first stage, a certain level of control on the development process may allow the definition of an architecture which will integrate physiological measurement devices. As for the last two stages, the architecture of the simulator is not always designed for human factors needs, thus seamless integration of eye trackers in the architecture is challenging. Usually in these cases, eye trackers will be used independently as standalone positions with their recording software and data analysis. Correlation with events in the simulation will require fastidious post-processing work. Despite these limitations, we explain how it is possible to facilitate the integration of eye trackers in complex systems by the choice of the software architecture.

2.1 Eye Tracker Systems Overview

Various systems are used to track eye movements [7, 11]. The setup can be head-mounted, table-based, or remote. These devices use video-cameras and processing software to compute the gaze position from the pupil/corneal reflection of an infra-red emissive source. To increase data accuracy with table devices, it is possible to limit head movement with a fixed chin on the table. A detailed description of the experimentation setup with the apparatus, the screen(s), and the subject is mandatory. A calibration process is also mandatory to insure system accuracy. The calibration process usually consists of displaying several points in different locations of the viewing scene; the ET software will compute a transformation that processes pupil position and head location [9]. Table-based eye trackers are usually binocular and thus can calculate eye divergence and output raw coordinates of the Gaze Intersection Point (GIP) in x-y pixels applied to a screen in real-time. This feature allows integration of gaze position as an input for the HMI. Areas of Interest (AOIs) are then defined to interact with the user. When the gaze meets an AOI an event is generated and a specific piece of information will be sent. When an AOI is

an element of the interface with some degree of freedom (a scrollbar, for instance), one is talking about a dynamic AOI (dAOI). Tracking of a dAOI is more challenging compared to a static one. Recently, Jambon [13] proposed a software architecture which allows the detection of a fixation in a dAOI. In this paper we propose a further generalization of this architecture. Another study proposed a tool for dynamic detection of AOI on a video or an animation [17].

There are two kinds of ET data collection methods. The first and the most common is to use the original software (for data recording and analysis) that is often provided by the device manufacturer. The second is to develop a specific software module (using a System Developer Kit (SDK), usually provided with the eye tracker) for data collection. Various parameters will impact the precision of raw data issued from the ET system. Among them, the video frame rate and the camera resolution are critical for the ET software. Existing systems use a video frame rate from 30 to 2000 Hz. For high precision ET, high frequency rate will improve data filtering but will also increase the data size and processing time which is critical for online processing.

2.2 Eye Tracker Data Analysis and Visualization

ET data collected during an experiment can be analyzed by statistical methods and visualization techniques to reveal characteristics of eye movements (fixations, hot spots, saccades, and scanpaths). A recent survey presents an overview of visualization techniques for ET data and describes their functionality [2]. Eye tracker data can either be processed offline for analysis purposes or online in order to adapt the HMI dynamically or to use gaze as a pointing device. Fixation, saccade, and smooth pursuit events [21] can be computed from raw data coordinates. To correlate these pieces of information with the HMI, some interface-related data have to be collected (i.e. object coordinates within the interface, HMI events like mouse hover, etc.). This information can be used to infer the user behavior:

- fixation (smooth pursuit) indicates visual encoding during overt orienting [8, 21];
- saccade is the process of visual research when the focus of attention is shifted;
- number of fixations on a specific object is often an indicator of the importance attached to a specific object [19];
- mean fixation duration or total dwell time can be correlated to the visual demand induces by the design of an object [14] or the associated task engagement.

Saccades are rapid eye movements that serve to change the point of fixation, and during which, as it is often considered, no information is encoded. Fixations occurs when the user fixate an object (usually during a 150 ms threshold) and encode relevant information. Sometimes shorter fixations are taken into account. Unlike long fixations that are considered to be a part of top-down visual processing, short ones are regarded as a part of bottom-up process. It is estimated that 90 % of

viewing time is dedicated to fixations [7]. Other complex ocular events like glissades or retro-saccades could be considered. There exist numerous algorithms of eye movement event detection [11, 16]. Still, there is no general standard for these algorithms. The integration of such algorithms will be discussed in Sect. 4. The blink duration and frequency can be used to assess cognitive workload [3], both of which can be collected with an eyetracker. Variation of the pupil diameter can also be used as an indication of the cognitive workload [15, 18], defined by Beatty [1] as task-evoked pupillary response (TEPR). However, light sources (environment, electronic displays, etc.) must be strictly controlled since the pupil light reflex is more pronounced than the impact of the cognition on pupil size. Moreover, even the luminance of the fixation area (even when the luminance of the computer screen does not change) has an impact on the pupil size. Scanpaths can also provide insight on HMI usage. In general, collected and cleaned data can be analyzed to infer causal links, statistics, and user behavior [10].

3   Design Rationale

In this section, we present our taxonomy of tasks and a structured design space where every need regarding eye tracker data record and processing is gathered. This design rationale is the result of two brainstorming sessions with four human factors experts with expertise in ET, a research engineer, and one researcher in HMI and Information visualization. This section will not provide architecture solutions but rather questions that will help users to correctly define ET integration requirements:

What to Record?

ET systems can record many ocular features, all of which will be merged for future analysis. These features usually include gaze position, pupil size, head position and movements, eye divergence (in the case of binocular system). Some higher level complex features like scanpaths or activation of AOIs can be computed and recorded. A more detailed description of recording parameters can be found in [7]. In the case of a head mounted device, a reference image is often used and recorded to compute head movements and to correlate gaze and head position on the scene. In other cases, there is no need to know where the participant is looking at, therefore no correlation is necessary (e.g., if ET is used to estimate the cognitive load and/or attentional state by eye ballistic or pupil diameter). Additionally, contextual information must be (other physiological data, HMI events). Researchers usually record every piece of information available in the case of later analysis unanticipated at the moment of protocol development. Still, increasing the number of recorded features decreases the processing speed and requires extra storage space. If the ET system is used as a part of HMI, then the processing time becomes critical for (near) real-time interactions. In any case, recording unused parameters is undesirable. Thus, the question of relevant ET features choice is important.

What are the Environment Constraints?

Experiments can be conducted in various indoor (laboratory or simulator) or outdoor (airplane cockpit, car) environments. Environmental constraints directly define the possible ET features regarding the research objective. For example, ET usage can be an issue in high luminosity environments. Constricted pupils of 2–3 mm in diameter are difficult to track. If the ET is head-mounted and uses a field view camera, excessive light can saturate the image such that head movement calculation is be impossible. While recording during real flight [7], pilots usually wear sunglasses so that the tracking becomes quite delicate. Other problems such as vibration or sun power can occur in these environments. With remote ET devices, body and head movements are also a critical issue for accurate data recording; eyes can be lost with excessive head movement. During the calibration, the experimenter should verify that the participant's head has some degree of freedom but a slight head/body movement would not cause a loss of ET. If the data has to be of extreme precision (e.g., for use in psycho-physiological analysis), a chin rest is recommended. When users must be able to freely move their head (e.g., multi-display environment), head-mounted devices are preferable. Some environments have safety concerns; in such cases, the ET integration can be delicate and a specific care must be taken to monitor gaze without compromising safety. This concern is especially true with airline pilots. Little experimentation was done, and currently most of them appeared in training simulators. Even in a training simulator, some constrains remain. Simulation sessions are costly and participants, particularly professionals, have limited availability. The ET setup time must be as short as possible. The chosen ET system (head-/table-mounted, or remote using multiple cameras) is also constrained by the environment. Head-mounted devices lose tracking less often because of the proximity of the camera to the pupil. Still, head movements could have an impact on data recording, if the position of the device changes considerably due to participants knitting the brows, for example. In the latter case, the device would erroneously map the pupil position on the field camera. The head-mounted ET is also more intrusive and could interfere with participant activity. Calibration can require some time compared to other types, and light context and the physical environment can hinder the process. In the case of a head-mounted ET, a special care has to be taken for correct installation to avoid device shift and re-calibration. In the case of a head-mounted system, head motion tracking can be improved with physical markers (e.g. AR marker). Some calibration techniques can take into account multilayer depth (e.g. modern glass cockpit usage). Calibration can be problematic when the subject is wearing eyeglasses or contact lenses, or if there is a reflection in the eye, or if parts of the pupil are hidden by eyelashes or eyelids. Globally it takes more time to calibrate and setup head mounted devices than fixed one.

What are the Recording Requirements?

Sampling frequency should be decided with caution regarding the usage of the eye tracker. In the case of accurate time frame, the amount of recorded data can be huge. In this sense, special care must be taken to handle this information in term of size and extra computation requirement can be especially an issue when several eye

trackers must be synchronized. Sampling frequency can be high for physiological experimentation, but a low sampling rate can be sufficient when doing general interface usage monitoring. In the case of complex experimentation platforms with multiple computers, time synchronization is crucial when correlation between various sources of data has to be made. For instance, if electroencephalography is used with an eye tracker during the experimentation, time synchronization has to be accurate to the millisecond. Finally, one must decide on the recorded data format. Usually human factors specialists will prefer comma-separated value (csv) files in order to process data in spreadsheet, statistical, or mathematical software. In case of high sampling rate like 500 Hz, file size can be an issue for text files. In order to assess the correctness of recorded information, one can monitor them with dedicated supervision of the HMI. For instance, camera accuracy, calibration quality, reference image, gaze position, reference image, and dynamic AOI all can be dynamically assessed. In case of incorrect data recording, it is possible to stop the experiment and to fix the problem. Supervision can also be used as a debugging tool, to retrieve a broad idea of the current experimentation validation. As a matter of fact, all recorded data can be monitored in real-time, but it can represent a technical challenge to visualize it.

What are the Processing Requirements?

Finally, processing and analyzing collected data can be time consuming, with important memory consumption and especially with big datasets. In some cases, real-time processing is necessary if ET is used as an input. Off-line processing can be used to manually adjust AOIs and to assess experiment results. Merging from different sources and data cleaning is always time consuming and is error prone. Thanks to the supervision, it is possible to qualitatively assess the recorded data. Data processing occurs with the mix of data cleaning and fusion, then finally, data can be segmented, processed and summarized.

4 Aeronautical Use Case

Contributors of this research paper come from the aeronautical domain and are human factors specialists and researchers in human computer interaction. We present a Stage 3 use case and the technical challenges addressed.

The main objective of this experiment was to verify in a complex setup the efficiency of a monitoring agent dedicated to assisting the controller. Three designs were selected for their saliency properties to be integrated in the radar image of a complex air traffic control simulator. This simulator is modular and constructed over the Ivy middleware [4]. Relevant information was detected by the monitoring agent and notified dynamically with the appropriate saliency on the radar image. A table-based eye tracker was chosen to measure the perception of the notifications.

Two technical challenges had to be addressed in this study: the integration of the eye tracker in this complex simulation platform and the complexity of the radar image; and the difficulty associated with the integration of ET data in complex software for the detection of dAOIs.

A specific module for fixation detection had been created to address the first challenge. Depending on the sampling frequency, it is possible to have a fixation module with a different algorithm without modifying the other modules. In this type of architecture, each function is a dedicated software agent in order to allow better flexibility in future experimentation. With our architecture, it is possible to add/remove a module depending on the study objectives, record relevant data for further analysis and even integrate high-level data like the workload index assessed dynamically by a mental state classifier. The architecture simplifies the post-treatment and the analysis phases for human factors specialists and allows HMI adaptation with ET data since raw data and high-level data can be collected dynamically on the software bus. Other optional modules could be integrated in this architecture like a dAOI module to correlate objects coordinates and fixation events or a TEPR module which analyze pupil data from the ET gateway and can produce outputs for a mental state classifier.

For the second challenge, two technical choices were possible. The first one was to improve he radar image code in order to output coordinates for meaningful objects and associate them in a dAOI module. The second one was to integrate fixations in the radar image to correlate them to objects inside the process. We chose the latter as our radar image was designed like Mozilla Firefox to integrate plugins to allow agile development of new functionalities without modifying the core application (dashed box in Fig. 1). Various plugins have already been created to support the use of a dynamic eye tracker like a gaze piloted cursor, an alarm validation plugin based on long fixations on notified objects, and detection of fixations on aircraft. Relevant information like graphical/data objects fixations were sent on the software bus and logged by a dedicated software agent. If the aim of the experiment is not to adapt the HMI but only record and correlate data between the eye tracker and HMI events or objects, an architecture without plugins and additional modules is more useful for reusability.

For this specific experiment, we have used the fixation module and a plugin included in the radar image for the detection of dAOI fixations and correlation with aircraft. Logging was done by a specific module and information send on Ivy (Fig. 1).

What to Record? Raw data and fixations on dAOI.

What are the Environmental Constraints? Controllers work indoors on a fixed screen; a table-based eye tracker is the better and simpler choice.

What are the Recording Requirements? In this simulator, we use a multi-computer architecture, as time synchronization NTP (Network Time Protocol) is mandatory. For data collection we used Ivy and a logging module.

What are the Processing Requirements? Detect and record fixations on aircraft (specific dAOI) to simplify the correlation process with events (HMI or monitoring).
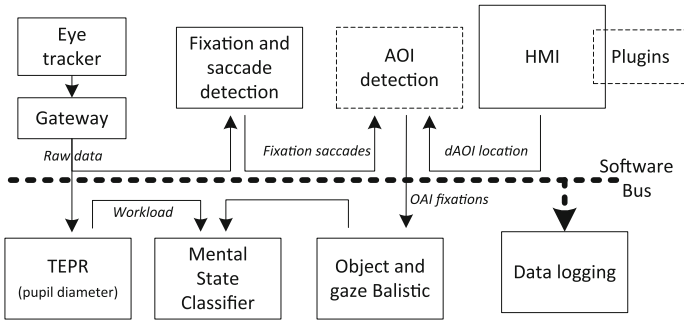
Fig. 1 A flexible and generic architecture

5 Conclusion

As a result, we proposed the following architecture recommendations:

- Use a software bus (middleware) to allow a simple and modular architecture and NTP protocol to address synchronization issues.
- Even in the case of laboratory evaluation (Stages 1 and 2 of the CSE framework) where research engineers have full control on the source code, using the more flexible architecture (use case 2) is the most relevant since it provides reusable components for future experimentation.
- It is mandatory to separate identified functions in a single software component and to connect them via a software bus. For instance, a fixation detection algorithm from raw coordinates or the measurement of TEPR (task-evoked pupillary response) amplitude can be considered as an optional and reusable module for future experimentation.
- The gateway between the eye tracker and the software architecture only outputs raw data from the eye tracker processed by another software component; no processing should be made. This allows for replacement of an eye tracker model with another by simply changing the eye tracker gateway.
- Raw coordinates coming from the eye tracker gateway should be stored in the log file even if high level eye-tracking data are already stored, since they can be replayed later for analysis purposes or processed by new filtering algorithms.
- Online calculation and correlations with HMI events must be performed when it is possible in order to store high level data (i.e. fixations on objects) and simplify the post-treatment phase for human factors specialists.
- Creating plugins for the HMI allows easier integration of high level ET data for HMI adaptability and permits a direct correlation between fixations and graphical objects.

This paper proposes a design rationale and architecture recommendations for their integration in the different stages of the CSE framework. These recommendations aim to improve the recording and analysis process for human factors specialist and

making possible the real-time use of high level ET data. A high fidelity simulation platform (stage 3) which uses this design rationale has been described.

The technical challenges have been addressed with a modular architecture and several reusable software modules created. Although we have successfully used this architecture for Stages 1 and 2, it is not well-suited for Stage 4 since it is difficult to gather data from a closed system.

To summarize our contributions, this paper provides taxonomy of ET usages and requirements for the software architecture. Although in the use case provided come from aeronautics domain, these choices are relevant for general HMI evaluation. This work is a first attempt to structure and rationalize ET integration in simple or complex simulation platforms.

References

1. Beatty, J.: Task-evoked pupillary responses, processing load, and the structure of processing resources. Psychol. Bull. **91**, 276–292 (1982)
2. Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., Ertl, T.: State-of-the-art of visualization for eye tracking data. In: Proceedings of EuroVis (2014)
3. Bruneau, D., Sasse, M.A. McCarthy, J.D.: The eyes never lie: the use of eye tracking data in HCI research. In: Proceedings of the CHI, vol. 2, p. 25 (2002)
4. Buisson et al.: Ivy: un bus logiciel au service du développement de prototypes de systèmes interactifs. In: IHM'02 (2002)
5. Bulling, A., Dachselt, R., Duchowski, A., Jacob, R., Stellmach, S., Sundstedt, V.: Gaze interaction in the post-WIMP world. In: CHI'12 Extended Abstracts on Human Factors in Computing Systems, pp. 1221–1224. ACM (2012)
6. Dehais, F., Causse, M., Pastor, J.: Embedded eye tracker in a real aircraft: new perspectives on pilot/aircraft interaction monitoring. In Proceedings from the 3rd International Conference on Research in Air Transportation. Federal Aviation Administration. Fairfax, USA (2008)
7. Duchowski, A.: Eye Tracking Methodology: Theory and Practice, vol. 373. Springer Science & Business Media (2007)
8. Goldberg, J.H., Kotval, X.P.: Eye movement-based evaluation of the computer interface. In: Kumar, S. (ed.) Advances in Occupational Ergonomics and Safety, pp. 529–532. ISO press, Amsterdam (1998)
9. Goldberg, H.J., Wichansky, A.M.: Eye tracking in usability evaluation: a practitioner's guide. In: Hyöna, J., Radach, R., Deubel, H., (Eds.), The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research, pp. 573–605. Amsterdam, Elsevier (2003)
10. Holmqvist, K., Andrà, C., Lindström, P., Arzarello, F., Ferrara, F., Robutti, O., Sabena, C.: A method for quantifying focused versus overview behavior in AOI sequences. Behav. Res. Meth. **43**(4), 987–998 (2011)
11. Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., Van de Weijer, J.: Eye Tracking: A Comprehensive Guide to Methods and Measures. Oxford University Press (2011)
12. Imbert J-P, Hodgetts H.M., Parise R., Vachon F., Tremblay S.: The LABY microworld: À Platform for Research, Training and System Engineering, HFES'2014
13. Jambon, F., Vanda L.: Analyse oculométrique « on-line » avec zones d'intérêt dynamiques: application aux environnements d'apprentissage sur simulateur. In: Proceedings of the 2012 Conference on Ergonomie et Interaction homme-machine (2012)

14. Just, M.A., Carpenter, P.A.: Eye fixations and cognitive processes. Cogn. Psychol. **8**, 441–480 (1976)
15. Matthews, G., Middleton, W., Gilmartin, B., Bullimore, M.A.: Pupillary diameter and cognitive load. J Psychophysiol. **5**, 265–271 (1991)
16. Nyström, M., Holmqvist, K.: An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. Behav. Res. Meth. **42**(1), 188–204 (2010)
17. Papenmeier, F., Huff, M.: DynAOI: a tool for matching eye-movement data with dynamic areas of interest in animations and movies. Behav. Res. Meth. **42**(1), 179–187 (2010)
18. Pomplun, M., Sunkara, S.: Pupil dilation as an indicator of cognitive workload in human-computer interaction. In: Human-Centred Computing: Cognitive, Social, and Ergonomic Aspects. HCII 2003, pp. 542–546. Crete, Greece (2003)
19. Poole, A., Ball, L.J. Phillips, P.: In search of salience: a response time and eye movement analysis of bookmark recognition. In: Proceedings of HCI 2004 People and Computer XVIII, pp. 363–378. Springer, London (2004)
20. Rasmussen, J., Pejtersen, A.M., Goodstein, L.P.: Cognitive Systems Engineering. Wiley (1994)
21. Reimer, B., Sodhi, M.: Detecting eye movements in dynamic environments. Behav. Res. Meth. **38**, 667–682 (2006)