Integrated Sensors Platform *

Streamlining Quantitative Physiological Data Collection and Analysis

Harry X. Li[1][0000−0002−2288−6039], Vincent Mancuso[1], and Sarah McGuire[1]

Massachusetts Institute of Technology Lincoln Laboratory

Abstract. Human performance measurement for computer-based tasks is a critical need for assessing new capabilities and making inferences about their usability, utility and efficacy. Currently, many performance-based assessments rely on outcome-based measures and subjective evaluations. However, a quantitative method is needed to provide more fine-grained insight into performance. While there are some commercial solutions that integrate human sensors, in our exploration, these solutions have numerous limitations. We decided to build our own web application platform, the Integrated Sensors Platform (ISP), to collect, integrate, and fuse disparate human-based sensors, allowing users to link physio-behavioral outcomes to overall performance. Currently, we have integrated three sensors, two commercial and one custom, which together can help calculate various user metrics. While our focus has been using ISP to assess visualizations for cybersecurity, we hypothesize that it can have impact for both researchers and practitioners in the HCII community.

Keywords: Tool Evaluation · User-Centered Design · Physio-Behavioral Monitoring

1 Introduction

When developing a new tool or application, it has become a common standard to incorporate the perspective of the user into the development. This process, also known as User-Centered design [4], is based on an explicit understanding of the users, task environments, and is driven and refined through multiple iterations.

Throughout the process, designers ask questions about the usability (can the user interact with the tool), utility (does the system do something of value), and efficacy (how well does the tool support the user's goal). To be successful in this process, designers must incorporate the users throughout each phase of the design process: interviews and observations during requirements development, participatory design and feedback during the development, and iterative user testing during development and evaluation phases [1]. In this paper, we focus on the last phase and present a new tool we developed to enable researchers and practitioners to execute rigorous and quantitative evaluations based on physio-behavioral data.

2 Human-Centered Evaluations

Human-centered tool evaluations often rely on subjective evaluations, by the user or observers, and outcome-based measures. Surveys such as the System Usability Scale [5] allow assessors to quickly and easily understand a user's opinion on how well the tool functions and how well they were able to use it. Methods such as heuristic evaluation and cognitive walkthrough [3] allow a subject matter expert to walk through a system to identify issues with the usability of a system based on usability principles and expertise in the domain. Alternatively, evaluations can rely on behavioral and performance measurements. These measures rely on procedural-based metrics (e.g. response time, time spent in search, order of operations) as well as outcome-based metrics (e.g. task completion, accuracy). While these metrics provide insight into higher level processes and overall performance, they still exist one level above the higher-level cognitive functions that drive human behavior, understanding and decision making.

Within the research community, the emergent field of neuro-ergonomics [8] has focused on identifying physio-behavioral sensors and developing analyses to elicit cognitive states such as attention, cognitive workload, stress and fatigue. Research has shown that measures such as eye movement, heart rate, and even user activity can be used to make inferences on these constructs [9]. Traditionally this research has focused on the antecedents and dynamics of the various states, but as our understanding matures, there are new opportunities in applying this knowledge to inform the human-centered evaluation of tools. For example, Lukanov et al. [6] leveraged functional Near Infrared Spectroscopy (fNIRS) to assess mental workload during usability testing of a simple web form. Their findings showed that the fNIRS system provided insight into changes in mental workload, some of which were unexpected, and demonstrated a method for complimenting subjective responses with quantitative physiological data. While this approach is useful for webpages and other general use tools for a wide user base, these effects are even more magnified when discussing tools for operational users, such as air traffic controllers, cyber security analysts, pilots and military personnel.

The approach, methods and findings of Lukanov et al. demonstrate the enormous potential for these approaches for human-centered evaluations; however, the monetary costs and need for technical expertise limit the use of these methods within User-Centered Design. As of today, the cost and complexity of acquiring, integrating and analyzing the sensors and associated data makes applying these methods less practical for use in usability and tool evaluations. While there are some commercial solutions that work towards this goal, in our exploration, these solutions are costly and can be limiting. Most commercial solutions are locked into the company's ecosystem, requiring users to purchase separate compatible sensors, and not allowing them to use sensors they may already own, or that may be more appropriate to their research needs. The lack of physio-behavioral measurement and analysis remains a critical gap in human-centered tool evaluations, one which, if addressed, can further our ability to understand the impact of our design choices on more complex and nuanced human outcomes.

In this pursuit, we developed the Integrated Sensors Platform (ISP), a web application platform to help collect, integrate, and fuse disparate physio-behavioral sensors, to understand cognitive outcomes such as attention and workflow, and link to the overall performance of the users, while making it accessible and affordable to human-centered design researchers and practitioners. In the following sections we will document the overall architecture of the system, the design choices that were made, provide an overview of how it can be used to conduct a tool assessment, document how it can be leveraged by the wider community and discuss our future work.

3 Integrated Sensors Platform

The ISP is a tool aimed at integrating a multitude of commercial and experimental sensors. Built using open-source platforms, we have focused initially on low-cost sensors as a way to help lower the cost of entry for users. Additionally, we provided built-in integration and analysis capabilities to allow users who may not have a background in such analyses to benefit from the tool. In the following sections we present an overview of the system, including an initial set of sensors we selected, description of the architecture and walkthrough of the user interface.

3.1 Sensor Selection

To demonstrate our concept, we selected 3 minimally invasive sensors that cover a range of behavioral and physiological measures to integrate into the system. We did not want sensors that would impede an individual's normal movement by requiring them to be connected to extensive wires. We also did not want sensors that required long setup or calibration time, since the purpose of the ISP is to enable a wide range of users to quickly capture relevant measures. Also, the sensors had to either be internally developed or open-source so that they could be integrated within the platform.

Heart Rate Monitor and Eye Tracker Based on these criteria, we first chose to integrate a Mionix NAOS GQ [7] computer mouse with an integrated optical sensor for measuring heart rate which can provide a measure of workload. We also integrated a GazePoint GP3 [2] eye tracker for capturing gaze position and blinks. These two commercial sensors come with software that makes the sensor data available through web server interfaces. It is important to note that we chose these sensors for the purpose of demonstration, and not necessarily based on comparisons to other commercially available models.

Sensei Software Workflow Monitor Finally, we built and integrated a software workflow monitor called Sensei which records user actions in the browser and also captures periodic screenshots for associating mouse movements and eye tracking to portions of the computer screen.

Sensei is written in JavaScript and can be easily integrated into any browser-based website. It listens for user-initiated browser events (mouse movement, clicks, key presses, etc.) and can stream the data in real time or batches to a server via HTTP or WebSocket protocols. Moreover, Sensei can take periodic screenshots of a user's tab or entire screen, which can be used to record user workflow and tool usage, and link physiological data back to what was on the screen at the time of collection. We have integrated Sensei into the ISP for troubleshooting as well as into external web tools that we wanted to evaluate.

This demonstrates the advantage of our platform, which combines data collection from commercial and in-house sensors.

3.2 Architecture

The ISP system architecture is broken up into four components: the Sensors which collect the data (discussed in detail in the previous section), the Sensor Proxy which transmits data, the Client which displays the user interface, and the Server which stores the data (Fig. 1). The ISP Client is a website written in ReactJS, JavaScript and HTML 5 that can be opened from any modern browser and run independently from other software installed on the computer. It receives data from the Sensors and/or Sensor Proxy, then visualizes and performs analysis on the collected data.

The Sensor Proxy is written in NodeJS and acts as a middle man between the Sensor and the Client for Sensors that cannot be directly accessed within the browser. For example, the eye tracker makes its data available on a TCP socket, which is currently tricky for a browser to directly connect to. Instead, the Sensor Proxy connects to the eye tracker TCP socket, then proxies the data to the Client via a WebSocket. Similarly, an external Sensei instance running in another web client can send its data to the Sensor Proxy which passes it to the ISP Client.

Finally, the Server is written in NodeJS and saves the data from multiple concurrent Clients to a local SQLite3 database file. Every time an ISP Client starts a data collection session, the Server assigns that session an ID and labels all incoming sensor data with that session ID. Later, the Server can separate data from different sessions using the session ID. In the future, the local SQLite3 file could be changed to a database server if better performance were necessary.

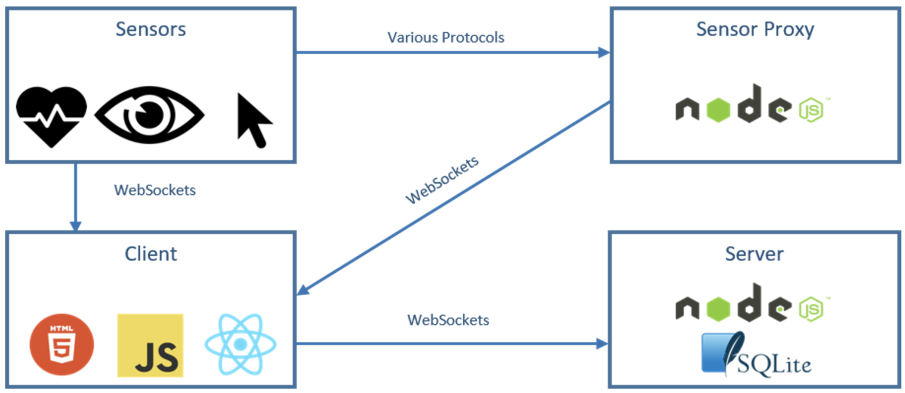

Fig. 1. Architecture Diagram of ISP. Users access the Client in a browser like a normal website. The Client connects to the Sensors directly or indirectly through the Sensor Proxy. The Client then sends data to the Server to be saved.

3.3 User Interface (Client)

The user interface has three primary pages, aimed at supporting data collection and analysis. First, the Configuration Page allows users to input their task information and configure their sensors. Then, the Data Collection Page lets user select sensors for data collection and displays the statuses of the sensors. Finally, the Data Analysis Page consolidates data to be downloaded and can automatically generate some analysis based on sensor data.

Configuration Page When first loaded, the ISP starts on the Configuration Page shown in Fig. 2, which allows the user to configure the data collection and record any necessary metadata. In the left column (Fig. 2 A), the user has the ability to save information about the current data collection session, including information about the task, participant, station, and equipment being used for data collection. In the right columns (Fig. 2 B), the user can configure the sensor data sources and where to save the data. The user can set the sample rate of the eye tracker (ex 30 Hz) and workflow monitor sensor, including the rate of the "mouse move" web event (ex 1 Hz), and configure how often to capture images of the user's screen (ex every 10 seconds). Finally, the user can configure the Client to collect data from a Sensor on a remote server by typing in the requisite URL or IP address.

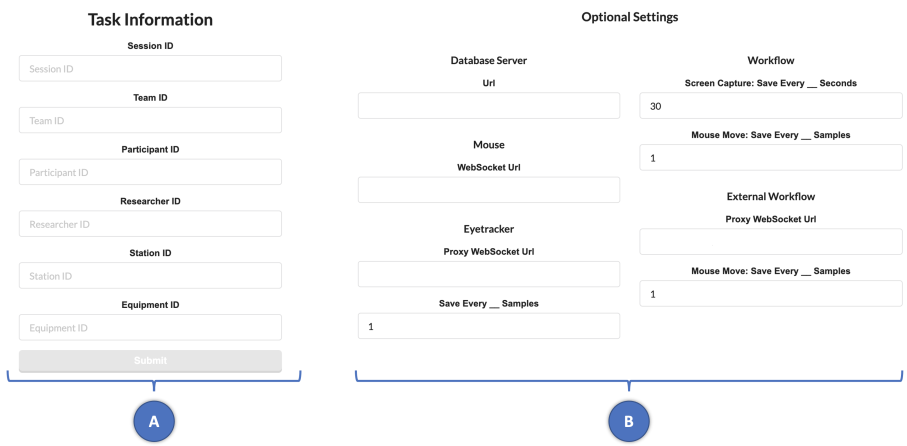

Fig. 2. Configuration Page. A) The user can input information about the task and B) configure how and where to save data from the sensors

Data Collection Page Once the user has entered the information in the Configuration Page, they can click the submit button to move to the Data Collection Page, shown in Fig. 3. At the top of the dashboard is a control bar (Fig. 3 A), that shows the task information and has buttons to start or stop data collection. Each of the columns shows information for the sensors currently integrated. Users can choose whether to save data from a sensor and are able to view the total number of data samples for that sensor. The left column is the heart rate mouse (Fig. 3 B), which displays the current heart rate. The next column is the eye tracker (Fig. 3 C), which displays the current and recent eye position data. The last two columns show data from a software sensor developed by our team called Sensei, which records web events (mouse movement, clicks, key presses, etc.) and takes periodic screenshots. The third column is the internal Sensei (Fig. 3 D), which displays the actions the user takes within the ISP Client and can be used for troubleshooting. The final column shows data from an external Sensei (Fig. 3 E) integrated into a separate web application, which displays user actions taken within that external tool. Once a sensor is enabled, data is streamed to the server in real-time. When data samples are saved from a sensor, it updates the number of total saved samples so that the user can monitor the data collection. The interface allows the user to control data collection and monitor the data being collected in real time so as to ensure accurate data is captured throughout session.

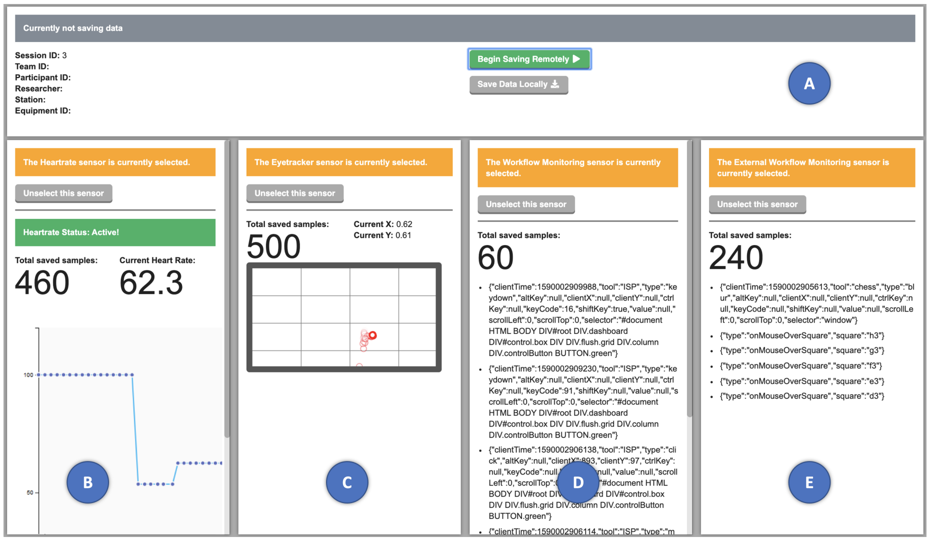

Fig. 3. Data Collection Page. A) The top control bar displays task information and has buttons to start and stop data collection. The bottom columns display sensor data and controls for the B) Heart Rate Monitor, C) Eye Tracker, D) Internal Sensei, E) External Sensei.

Data Analysis Page After data has been collected, the user can access the Data Analysis Page (Fig. 4) where they can access an archive of the data on the server and can perform some analysis. In the left column (Fig. 4 A), the user can configure where the Server is located. When the Client retrieves the data, the web page displays a list of all the data collection sessions stored in the database. The collections are sorted in order according to the datetime created and displays the task information that the user had entered in the Configuration Page at the beginning of data collection. When the user clicks on one of the tasks, the right column shows more information about the collected data (Fig. 4 B). The user also has the option of downloading all the collected data from an individual session or all the sessions in JSON format.

To correlate the mouse and/or eye position data to user workflows, Sensei takes periodic screenshots that the Data Analysis Page uses to generate heatmaps (Fig. 4 C). The user can select the source data (mouse or eye positions), and the Data Analysis page will generate heatmaps by downsampling and summing the position data into a lower resolution grid. As the screenshots are taken periodically, the heatmaps may not always accurately represent what the user was looking at in between the screenshot samples, so it is important that the user keep accurate records to ensure heatmap validity.

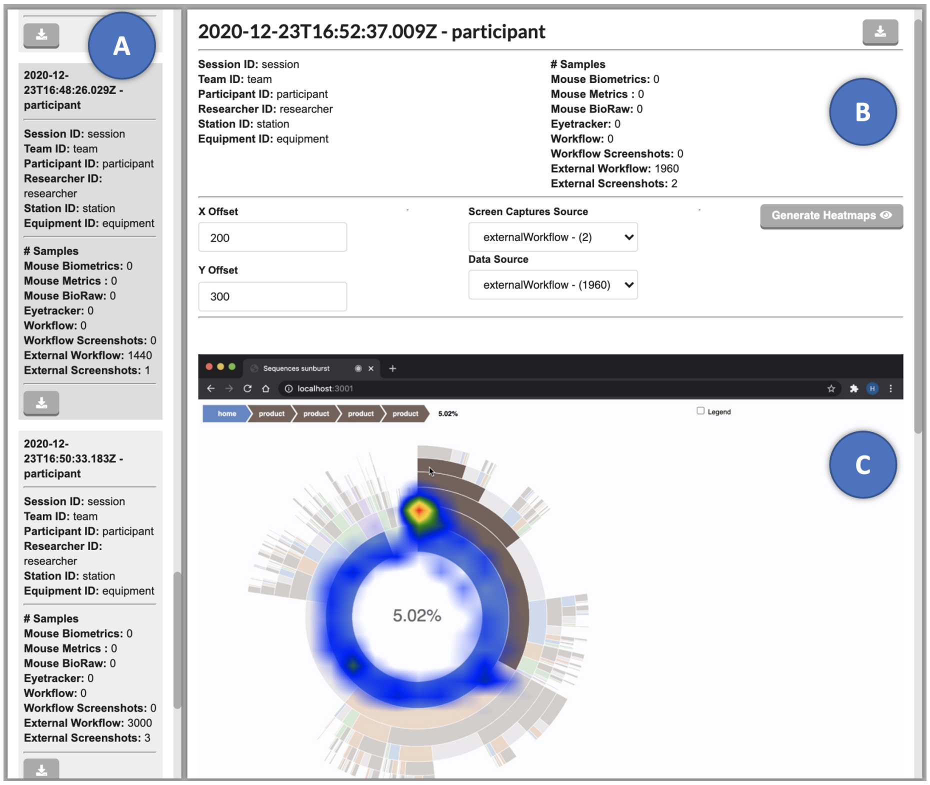

Fig. 4. Data Analysis Page. (A) list of data collection sessions, (B) more information about a particular session and (C) example heatmap generated by ISP based on usage of a sunburst visualization.

4 Applications of ISP

In developing the ISP, our main intent was to use it for evaluating analytic and visualization tools for cybersecurity. However, we hypothesize that there are more broad applications. By making a low cost, easy to use, and composable suite of tools, we aim to enable users, who may not have as rich of expertise, to leverage research on physio-behavioral monitoring in their human-centered evaluations.

ISP could be used, for example, to quantitatively evaluate the effectiveness of a computer operator or the usage of a prototype web tool. A researcher could conduct a study with ISP to collect data on participants as they perform computer tasks. Afterwards they could analyze the data to evaluate workload, fatigue, user workflow, or tool usage, then make quantitatively-backed recommendations to improve operator training or tool design.

Moving forward, our goal is to build on the ISP based on our own experience, and work with potential stakeholders to create a more holistic tool that enables users across multiple domains to understand physio-behavioral outcomes of their tools on users.

5 Future Work

As we continue to develop the ISP, we plan to expand the number and types of sensors into the system and expand the automatic analysis capability. There are several sensors that could be added to expand our current data collection ability, including a proximity sensor to measure collaboration, chat monitoring to measure communication, and fNIRS to improve the capture of cognitive load. We also hope to expand the capability to allow for different types of heart rate monitors (i.e. chest straps or wrist worn), and eye trackers. In addition, as remote human participant testing is increasing, capabilities for remotely capturing eye tracking and physiological measures using web cameras could be integrated into the system.

The automated analysis tool can also be greatly expanded to provide workload information from the heart rate and eye tracking data, and task performance measures could also be derived from Sensei-captured data, such as task completion time and tasks completed. Since our goal is for this system to be usable by both novices and experts in physio-behavioral data collection, better analysis capabilities could provide useful results for all experience levels.

Moreover, while we have currently only used ISP in small-scale testing and data collection performed by single team members, we would like to use ISP to capture data on multiple people collaborating, in order to test and demonstrate this capability. We also want to validate ease of set up: that the Sensors and Client can be set up on users' computers, and that the Sensor Proxy and Server can be set up on a server or virtual machine in the network. Finally, we want to validate that the Server is robust enough to concurrently collect data in a distributed environment, and that data collection is comprehensive and encompassing.

6 Conclusions

We developed the Integrated Sensor Platform to help users who are interested in evaluating tools, and conducting human-centered research to collect, integrate, and fuse data from multiple human-based sensors, and link them to factors such as stress, workflow, etc., and to the overall performance of the individuals and teams. We developed the platform with flexibility in mind, allowing for the addition of multiple types and brands of sensors to the system. By leveraging common, open-source web technologies, we designed a modular system that can be easy for developers to expand the capabilities based on their specific needs, available sensors and research questions. Our initial prototype system integrates three different sensors: a heart rate monitor that is built into a mouse, a low-cost eye tracking system, and an internally developed software workflow sensor called Sensei. In addition to flexibility in the code base, the ISP was developed to allow flexibility on the front end. The user interface allows individuals to select the sensors used and to monitor the data collection. Once collected they are able to easily access, download, and view initial analysis of the captured data. We have prototyped initial visualizations into our analysis system that will allow users without expertise in physio-behavioral data analysis to understand the data. By creating this system, we hope to reduce the barriers to collecting physio-behavioral measures making it more accessible to collect this data when performing usability assessments of technologies as they are developed or to capture quantitative measures of performance in research activities.

References

1. Abras, C., Maloney-Krichmar, D. and Preece, J. 2004. User-centered design. Berkshire Encyclopedia of Human-Computer Interaction. W. Bainbridge, ed. Berkshire Publishing Group. 445–456.
2. GP3 Eye Tracking device. Gazepoint. (2021, December 16). Retrieved January 5, 2022, from https://www.gazept.com/product/gazepoint-gp3-eye-tracker/
3. Hollingsed, T., Novick, D.G.: Usability Inspection Methods after 15 Years of Research and Practice. In: Proceedings of the 25th Annual ACM International Conference on Design of Communication. pp. 249–255 Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1297144.1297200.
4. International Organization for Standardization: ISO: 13407: Human-Centered Design Processes for Interactive Systems. International Organization for Standardization, Geneva (1999).
5. Lewis, J.R.: The System Usability Scale: Past, Present, and Future. null. 34, 7, 577–590 (2018). https://doi.org/10.1080/10447318.2018.1455307.
6. Lukanov, K. et al.: Using fNIRS in usability testing: understanding the effect of web form layout on mental workload. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. pp. 4011–4016 (2016).
7. Mionix Naos QG. Mionix NOAS GQ. (n.d.). Retrieved January 5, 2022, from https://mionix.net/products/naos-qg
8. Parasuraman, R.: Neuroergonomics: Research and practice. Theoretical issues in ergo-nomics science. 4, 1–2, 5–20 (2003).
9. Parasuraman, R., Wilson, G.F.: Putting the brain to work: Neuroergonomics past, present, and future. Human factors. 50, 3, 468–474 (2008).