

Chapter 10: User-Centric Design of Web Dashboards for Live Data Visualization and Control

10.1. Introduction

Users interact with more and more live data every day. They want to control the data shown and the way it is being shown. This motivation leads to the development of web dashboards for live data visualization and control that follow a user-centric approach. Live data can be retrieved from sensors and frequently changing data in web-based games. Examples of user control are moving sliders to control a game or selecting options to view differently aggregated data in charts. Principles of user-centric design cater to the user's needs during the development process.

To better understand the user, user research is carried out to gather user needs. The information guides the design process and forms user personas that represent the characteristics of typical users. Interactions between the user and the system determine the UI design of a web dashboard, describing the layout of the website in terms of charts, toolbars with buttons, sliders, drop-down menus, etc., and navigation within the page or to other pages. The UX describes the overall experience of the user when interacting with the interface, contributing to the creation of products that are easy to use, efficient, and enjoyable. Applying user-centric development principles in data visualization creates webpages that allow users to not only interact with the data but also truly control the representation of the data.

10.1.1. Overview and Objectives

User Centric Design (UCD) is a design process focused on users and their needs at each phase of the design process. User requirements, usability, and user satisfaction are some of the aspects enhanced by user-centric design processes. UCD processes have been applied to all types of software products and services, but also to non-information systems like ceramics and furniture. A Web dashboard is an information deliverer

through Web interfaces on desktop and/or mobile devices. It provides at-a-glance views of key performance indicators (KPI) relevant to a user or group. Based on a user-centric process, Web dashboards are designed, implemented, and eventually evaluated for the visualization of control parameters of a plant in near real time.

In recent years, a visual discrepancy has been observed in data representation. On one hand, the technological advances in information processing have led to highly sophisticated visualization tools, which are able to tackle complex massive datasets generated in almost every area of knowledge. On the other hand, many common users are still facing technological limitations concerning the decoding of data and require help to understand the information, using simple but appealing visual constructions. In this regard, Web dashboards are implemented focusing on near real-time live data. Live data is a term used to refer to an information flow that is configured in the system through a wizard by defining a data retrieval period of minutes, seconds, or hours. There exists an important variety of products, services, solutions, tools, and applications that offer visualization of information through Web interfaces on desktop and/or mobile devices. However, most such products and services have a fixed structure and do not provide individualized support for the user.

10.2. Understanding User Needs

The design of web dashboards for live data visualization and control generally follows the principles of user-centric design. The process broadly involves three stages. First, John et al. conducted a contextual inquiry with industry practitioners who deal with live data to understand their practices and needs. The interviews revealed that, while a variety of live data from multiple domains feed into their decision-making processes, these live feeds are juxtaposed by query, subsequent retrieval, and summarization of historical data [1-3]. It also revealed that decision makers make use of domain expertise and intuition. Second, based on the lived experience of the intent of live dashboards, two design workshops were conducted to collect feedback on existing and potential new live data visualization and control dashboards. Finally, the findings have been synthesized into design guidelines for dashboards that visualize and control live data.

Contextual Inquiry. Contextual inquiry is a technique to assess users' goals and tasks by studying how they interact with the environment. Industry practitioners dealing with live data were interviewed using the technique. The interviews addressed known data visualization design issues such as the difficulty of visual analysis on live data, the requirements for client-side versus server-side visualization, and the use of visualizations to make decisions and subsequent actions for which control widgets are required. A total of thirty-one participants were interviewed: six from social media, seven from construction, two from power system, two from transportation, two from

Telco, four from water utility, two from marketing, three from emergency services, and three from finance.

10.2.1. User Research Methods

In the many fields of HCI, relating the user to the interface remains an important part of the research process. Knowing the user is crucial to knowing the interface. It becomes a common approach before the design progresses much past the equivalent of a sketch of the interface to look at the design through the lens of the user.

An interaction design process says that the design needs to be grounded in a thorough understanding of those who will use the product. However, a quick look around the Web shows that most Web dashboards seem to be addressing anything but the needs of the user. User research was carried out on some key services currently provided to residents of Waterford City. These highlighted common problems with the services and dashboards, and also uncovered common requirements and preferences for the new dashboard. These lessons were then adapted and applied to the area of WSNs.



Fig 10 . 1 : User Research in HCI and Dashboard Design

10.2.2. Defining User Personas

To explore the influence that a specific control system has upon the design of its web dashboard, an investigation was undertaken to characterize the typical end users. Using

standard methodology, a questionnaire was circulated among university students and researchers. In total, 26 participants completed the survey, with 21 self-defining as researchers and 5 selected as students. The primary rationale for the diversity of these profiles is the different levels of interaction with the control system. While researchers generally monitor the behavior of the systems or modify their configuration, students use it to analyze live data.

The survey comprised seven questions that delved into diverse aspects of the control system. These included whether the respondents send commands or merely receive notifications, the purpose for employing the control system, the need to share data, and the type of data they monitor. An additional question asked what other systems or control dashboards the participants were familiar with. The results were synthesized to create two specific user personas: the Control Expert and the Control Beginner.

10.3. Design Principles for Dashboards

Design principles are discussed that integrate responsive and adaptive design methods to develop flexible, engaging, and user-centred digital dashboards. Requirements for viewing a digital dashboard across an increasing number of devices and platforms underpin these principles, when combined with the customisation demands of users from different sectors such as finance, energy management, aviation, retailing, and emergency services. Flexibility involves both responsiveness—automatic adjustment to different screen sizes and orientations—and adaptability, empowering users to apply personal changes. User-centred design is emphasized by enabling user control of the volume, layout, and presentation of data [3-5].

By combining responsivity and adaptivity, a user is supported in determining how much data to display at once (volume), where the information should appear (layout), and how it is presented (presentation). Movement of elements, resizing, reordering, and style adjustment simultaneously increase the levels of data shown, are managed with minimum scrolling, and provide a change of emphasis, respectively. Web-based desktop, tablet, and mobile digital dashboards are demonstrated in the domain of live grid demand-side frequency control in the national electricity system. User evaluation confirms that increased control leads to a better user experience.

10.3.1. Clarity and Simplicity

Clarity and simplicity are important in web design, especially on small screens. The display should present only the necessary data to avoid confusion and cognitive overload. Responsive views should be designed to prioritize the most relevant data,

ensuring succinct and digestible information. Interactive drilldown can help de-clutter pages by revealing additional details on demand. In the context of dashboards for massively connected sensor data, careful design of colors, contrasts, and sizes is crucial for easily distinguishable and low-effort visualization of information on any device.

In real-time dashboards for specific systems, the use of additional shades or the blinking of colors across containers or Table elements is commonly employed to indicate the operational status and raise alert messages. Conversely, the more journals and books a site has, the less chance the Journal Alerts system provides matches for the search terms. If an alert can't be created, a message indicating the absence of matching journals or books is displayed. Users are then encouraged to lessen the specificity of their search terms or adjust the results after an alert is set up.

10.3.2. Consistency in Design

Dashboards in a web development context can be defined as graphic displays used to monitor and control the schematics of complex systems. The graphic emphasis of these displays ensures that the information can be read and interpreted at a glance by the user, providing clear, concise, and intuitive data visualization. When the dashboard is dynamic and the information is consistently updated, these characteristics become even more crucial, as the data changes over time. The primary purpose of the dashboard is to offer a user-friendly interface that facilitates any control or setup action during the system's operation. Dynamic dashboards represent one of the most sophisticated implementations in web development, requiring an appropriate knowledge of the design methods and control strategies outlined in previous sections.

Before implementing any design, it is essential to consider the consistency requirements. A good user interface necessitates a consistent design, not only within an individual section of the dashboard but also throughout the entire website. The positioning of each element and its parametrization must align with its functionalities. Moreover, animations should not compromise the safety and clarity of the displayed data. Ideally, user actions are listed on the left side of the dashboard, followed by system-related information. The first column, corresponding to the readings of the implemented sensors and the system's instantaneous condition, provides a synopsis of the entire scheme. Consequently, the first sector where alarms are located is visibly centered on the page, given their critical importance for plant protection and their required reaction capacity.

10.3.3. Responsiveness and Adaptability

The Dashboard monitors data generated by a MESN control system. In MESN, the NUCs are represented as UGVs, and the data generated by the various sensors is transmitted to the control center through 3G/4G. Since the system architecture is client-server, a dashboard may be used by multiple users simultaneously, and users may use any device to access the interface.

Responsiveness is focused on adapting the data presentation to the device screen properties regardless of other screen properties. To achieve this goal, the Bootstrap framework, a web front-end development framework, was used. Bootstrap follows a mobile-first approach, so it provides a grid system, pre-built components, and powerful plugins based on jQuery that enable the automatic adaptation of web layouts at different screen sizes [2,4,5].

These features allow the use of a data grid to reposition, show, or hide the information elements, depending on the screen width. The map container is configured to take the available space according to the height and width of the user device. As a result, on mobile phones, only the list of UGVs is shown along with the currently selected UGV location. On tablets, the map is shown at the top of the screen.

10.4. Data Visualization Techniques

Visualizing STREAM results in a web dashboard requires combining different visualization techniques. Live data is displayed in views and widgets, which offer options for customization. Tabular data is presented in tables, real-time data is shown in line charts, and geospatial data is rendered on maps.

Line charts and maps support plotting of not only data on a single processing node but also entities on a processing path, such as the number of flow files processed, the processing rate, or the transfer rate. Each widget can be configured to show a specific data field. Organizations that operate the data flows, for example, through the STREAM user interface, can customize these views for their requirements using the provided options. Each widget supports various settings for selecting the displayed data and defining alert rules for the live data.

10.4.1. Choosing the Right Visuals

With a high volume of data to analyse, a well-designed dashboard gives a user the ability to drive their findings.

Dashboards monitor system data and provide users with on-demand, real-time scheduling information. Dashboards rely on a dynamic display of visuals—charts, maps,

or gauges—that maintain the proper level of detail for the users’ scheduling needs. An ideal dashboard enables a user to easily create their views—combining simple visualization techniques—using live data, fuel scheduling tools, and stored historical information. The focus is on control to adjust the flight schedule, with the ultimate goal being the ability to reschedule and refuel in the same place. These design objectives align with user-centric design, which focuses on users’ needs and expectations for a product or service. They differ from data-driven design, which attempts to craft a user interface that mirrors the characteristics of raw data, regardless of user needs. A dashboard that reflects the actual hierarchical structure of the data will likely provide intuitive drilling and sorting features, but users cannot be expected to incrementally learn the data’s internal operators’ semantics.



Fig 10 . 2 : Dashboard Design: User-Centric vs. Data-Driven

10.4.2. Interactive Data Elements

The scripting session of the Hololink web dashboard is designed to resemble everyday web pages. The data elements are organized according to the web page author’s design. When a user selects a data element, a live data stream from ROS is activated and sent from the Web API to update the element on the displayed web page. Multiple elements can be activated simultaneously, which can provide users with the capability to correlate cross-domain data for anomaly detection and monitoring.

Users can also enter new input values by double-clicking data elements that have the "input" attribute enabled. HTML forms that support single-value entries, such as Radio buttons or Text Areas, are supported so users can choose a single icon to trigger a scenario or input multiple parameters to control a robotic system. The Web API then publishes the new input values to ROS through the corresponding ROS topics. These interactive elements enable users to remotely control, trigger, or upload scripts to robotic systems from web browsers.

10.4.3. Color Theory in Visualization

Color visual features are widely used in infovis. An early pioneering theory (Boynton 1979) suggests that luminance is the key feature for detection, but hue and saturation are easier to remember and identify. Colors are explained in the wavelength dimension of the visible spectrum. Another approach focuses on the relation between colors and their psychological effect. Painting theories, such as that of van Gogh described by Winston (1995), stress the importance of complementary colors; the opponent theory of color vision proposed by Hering (1872) forms the basis for this effect. These views are also mentioned in minard.fr/infovis-color.

On the topic of color in the visualization of statistical data, Cohen (1987) analyzes perceptual and cultural aspects, concluding that color fully deserves its fame as a graphical variable [6-8]. He argues that it is perceived pre-attentively, with the human eye detecting small differences; that it enables data encoding for a broad range of variables, such as temperature, groups, and risk level; and that colors benefit from universal cultural associations.

10.5. User Interface (UI) Design

Cutting-edge web dashboards provide users with convenient access to relevant data. The design direction for the current dashboard is the systematic, application-appropriate, harmonious, and restrained use of materials for UI components. Materialized design strives to intelligently process Web page components, making content presentation more reasonable, detailed, and embodied, thus enhancing users' ability to concentrate on the content itself rather than being disrupted by beautiful yet distracting UI components. The dashboard's articulation encompasses three UI architecture layers: Material, Structure, and Skin. Material-coded UI elements deliver a sense of materialization; Structure introduces styles and layout of UI elements to the page; and Skin shapes the style of each component in each state. Rhythm integration comprises several key aspects: the selection of materials (such as black or white) for each component, the size and shape of component graphics, color matching, size matching, and layout.

The Support System Integrator (SSI) assists system integrators in managing an extensive range of physical devices deployed throughout the organization, in addition to managing users. It includes functionality to acquire data from physical devices, manage related services, and configure alarms. SI features comprise device management and definition, user group configuration, service management, and alarm rule configuration. User and User-Group Management, as well as Alarm Acquisition and Dispatch modules, are shared services across Systems and Services. Physical Device Management allows the creation, deletion, and modification of physical devices, as well as querying and configuring device attributes. Alarm Rule Configuration defines the monitoring range of devices. Service Management maintains services in the systems and services framework, enabling control, suspension, or closure of services as needed. Device Definition includes setting Public Database fields, Device Service fields accessible to the Outside World interface, and mapping Public Device Information to service device fields to support Service-Device Operations.

10.5.1. Layout and Structure

Information dashboards visualize large volumes of data and are useful for handling and controlling complex systems. Live dashboards display metrics and provide real-time system control. Web dashboards can be complex; a grouped system removes redundancy and concentrates common demand and control in one site. A set of dashboards that show the current values of a group of devices and allow data modification and the introduction of reference data has been developed. The solution allows the appearance of the website, sections to be displayed, data visualization, and control at different levels of detail to be customized. Python Dash was used to implement the software, extended with the Dash Bootstrap Components library.

The groups of sensors chosen for this case study reflect a user demand of the Control and Environment Monitoring Section of the UPC Building Technical Services. The data groups have different frequencies of information update, from 5 s for weather information and indoor thermal conditions to a daily update for the electricity price. The current requirements for each data group establish which content must be displayed in the dropdown menu and the customization of the sections in the detail view of the groups. The user demand engaged in the development of the dashboards provides insight into the functioning of the technical services and will facilitate the use of the presented software by staff and tenants of the campus.

10.5.2. Navigation Design

Navigation is an important element for Web dashboards. When users migrate from other dashboards, they want some similarity with their accustomed behaviour. Navigation on these dashboards is usually divided into panels and buttons. At the top of the screen, there is a header with information about the project or company. Although it is not a hard rule, links to documentation, forums, and management may be located there or in the lateral panel.

The sides of the dashboard's workspace may contain context or navigation panels that can be configured to stay on all the pages. Buttons are often placed in the context panel to create links to other pages. The active page is usually represented in the context panel, the header, or both. Navigation links can also be placed on the pages themselves. No hard rule exists to do so, but links pointing to one page in many others can be repetitive and bothersome.

10.5.3. Accessibility Considerations

Information and visual analytics projects deal with graphically synthesizing and representing information, with the aim of generating reports containing the most important conclusions. In this phase of the project, it is essential to focus on the final report, giving the user access to the information needed, focusing on the citizen who will consult the available information. Once the information has been extracted and the key points identified, the final phase centers on graphically representing these results. This section describes how the data, by lines of business or by registrations, is VIANAI's public sources. To guarantee information accessibility, the report is hosted on a public web server, making data and graphical representations available and facilitating decision-making.

The COVID-19 pandemic prompted numerous reports to assist decision-makers. These reports, summarizing large datasets, are prepared weekly or monthly. The conceptual design of a system aims to provide updated information through a web-based report. To be functional, reports require an infrastructure and database that updates information automatically and allows interaction with the graphical elements. A live dashboard responds to these requirements. Additionally, the system can be consulted using various devices and is supported by an intuitive graph and control menu system. Dynamic graphs enable the addition of elements to enhance data analysis and interpretation. Finally, support for information download and a user guide is included.

10.6. User Experience (UX) Evaluation

Incorporating the user perspective is vital when designing any solution, but especially when referencing web-based dashboards with specific visualization and control features. The user experience (UX) is generally evaluated for determining the efficiency, satisfaction, and ease of use of a system or device. UX evaluation can be classified into three approaches: expert evaluation, heuristic evaluation method, and empirical evaluation. Experts or human factors specialists access the system and conduct the expert evaluation without connecting to potential users. The heuristic evaluation method requires human experts with high experience and expertise. Empirical evaluation requires the solution to be tested on the potential users and gathers feedback on the identification of potential problems.

Several studies present UX evaluations applying these three different approaches. Yuen et al. apply a survey among potential end users to evaluate the interaction and the quality of the augmented reality (AR) applications. Qin et al. propose an AR VR-based mode to visualize the indoor environment, and also present a questionnaire-based survey to gather feedback from the users and, thereby, validate the impact of the method.

10.6.1. Usability Testing Methods

In the domain of software engineering, the assessment of the interactive system's abilities, ease of learning, efficiency, memorability, user perception of comfort, and error rate is referred to as a usability test. Measuring how well an application behaves is considered a key factor in making the system more popular to end-users. Numerous studies on usability testing have been conducted, addressing diverse application types, specific groups of users, tasks, and the application of various testing methods. For example, usability testing of open-source GUI applications can be performed using a questionnaire-based method. Another distinct area of investigation is the usability testing of smartphone-based mobile applications, where the degree of user satisfaction—related to ease of use, aesthetics, and learnability—and the application's functionality are evaluated. Yet another type involves usability testing of Web applications by means of heuristic evaluation, which can be effectively applied before launching an application in order to find major problems associated with the Web application's usability. Usability testing has also been applied in domains such as Alberta Government Websites and SMS, as well as in mobile commerce [7,9-10].

All these instances demonstrate that usability tests are not limited to just one specific kind of application or a particular demographic of users. The design and implementation of a particular task environment can also be tailored according to the area to be tested and the users involved, along with their specific tasks. Accordingly, for the

implementation of a live alert system and real-time updates, which primarily involves communication using emails, the usability test of an email service can be conducted by posing several tasks related to composing, sending, saving drafts, printing, working with contacts, and so on. The tasks that the user has to perform are prepared in such a way that all related tasks belonging to a particular set constitute one group. An interface is then developed, organized into different pages according to these task groups. On each page, a list of tasks for the user is provided in a panel on the left side.

10.6.2. Gathering User Feedback

The realized dashboard allows for the easy addition of more dashboards, with which it is desired by any means of visualization—not necessarily graphs—provided that there is access to the processed data to be sent to the WebSocket to be shown. Additionally, reverse communication with the server system is possible, i.e., the sending of control information to the source. In the implemented case, bidirectional communication is configured so that the IDs of the possible modifications in the lamps can be sent; however, only those corresponding to the control of the lamp are currently operative. With this scheme, the configuration of other operations of the video lamps is enabled, such as brightness, light modes, and colors. Thus, the most important aspect provided by these implementations is flexibility and scalability.



Fig 10 . 3 : Web-Based Dashboards for Live Data Visualization and Control

The proposed Web-based Dashboards for Live Data Visualization and Control enable remote control and monitoring of the video lamps connected to the WebSocket server. They make use of websockets to start and maintain communication with the processed

data server, allowing the exploration of the present data through various visualization techniques, such as the use of line graphs for temperature and humidity variables. Each of these dashboards can be accessed from a desktop or mobile device and can analyze other types of data for which the provided visualizations are applicable, whether at the explored moment or over the entire processing time window. The responsiveness of the Web application allows users to work with the dashboards on any device.

10.6.3. Iterative Design Process

Choosing a suitable technology for developing a Web GUI is almost always a trade-off between development speed and other important criteria such as simplicity of deployment, interoperability, graphical performance, control precision, or ease of maintenance. In the iterative design process, JS/HTML5 are among the approaches that can provide a high historical development speed. A large number of third-party JavaScript libraries can be freely used through the npm package manager, thus significantly reducing the time required to implement additional features, such as charting or internationalization. Moreover, a much larger number of developers have experience with Web development than with software for embedded devices. Although this background logic can be written in various languages, the logical choice is NodeJS, which is based on Google's V8 JavaScript engine. In particular, the easy setup of HTTP and WebSocket servers, along with a decent real-time performance, is an important advantage. It is also convenient that the developer can use the same language on the back-end and the front-end.

The downside of using JS/HTML5 is the lack of GUI elements and styling possibilities necessary for Web dashboards with only a simple layout. Instead of a row or a column, a div can be placed at an arbitrary position on a Web page. However, the size proportions of such a layout are difficult to maintain. If many widgets are placed as absolute-positioned elements and adapted for minimal and maximal dimensions, the maintenance of their layout may become more complicated than for more sophisticated layouts. A lack of styling possibilities can lead to graphical elements being used for non-graphical purposes. An example of this is the use of dials and sliders as state indicators due to the absence of switch and indicator light widgets. With appropriate attention to detail, a pivoted indicator may be a better representation of data compared with just a Boolean light. It can also provide the ability to intuitively anticipate a future state of the switch or indicator light.

10.7. Future Trends in Dashboard Design

The Web environment provokes questions about the implications and impact of wireframes and prototyping, widgets, and the REST architectural styles. Wireframe prototyping makes it possible to design Web pages that take into account Ajax interactions in the early phases of interaction design. Most existing toolkits and JavaScript widget libraries have been designed to provide very basic interaction style implementations. Dashboards have not been designed for REST, and the latter helps to identify many requirements for the Widget Pattern in modern Web apps.

Dashboard design tends to remove all unnecessary details and control elements that may create distractions or partiality. Researchers systematically investigate user data-management preferences and cognitive needs to then create a tool enabling users to design and build personalized Web dashboards that satisfy these needs.

10.7.1. AI and Machine Learning Integration

Machine learning agents offer an enhanced user experience for data visualization and control dashboards. They can provide instructions, demonstrate capabilities, and guide novice users through interacting with complex live data. Machine learning has been proposed as a method to predict dynamic energy management for the smart grid in energy companies. Such functionalities can be implemented on the server-side and exposed through an API that the dashboard can consume.

Machine learning techniques are also used for image recognition and classification. For dashboard monitoring of industrial sites or the environment through computer vision cameras and drones, the same process can be applied. The output of the classification algorithm would depend on context and use case—for example, visual intrusion or wildfire alert. Alert notifications can be generated based on the linked machine learning predictions.

10.7.2. Augmented and Virtual Reality Applications

The concept of using Augmented Reality (AR) and Virtual Reality (VR) for an intuitive display and control of data is already decades old. With the recent developments in the fields of AR and VR, however, new possibilities for such applications are emerging.

A current AR/VR use case is the positioning of fixed elements. Since planners are unable to perfectly imagine what the planned project will look like, simple sketches are used to describe the plan to customers, the public, or even within the planning team. Developers now use AR glasses to support the visualization of the project at the actual position,

thereby providing customers, the public, and other planners a better understanding of the final result through the entire planning process. AR and VR have proven to be helpful for more immersive visits to exhibition spaces, industrial buildings, and museums, even before they open. Some VR applications allow users to view the Earth at any altitude and position in a virtual environment. Visualizing position, altitude, and flight path helps both users and developers to create a better flight experience for night flights. AR's gyroscopic functions enable users to discover sights around them and receive further information easily on their own. A further idea is spending the winter months underwater in an aquarium to relax with nature while city noises disappear.

10.8. Conclusion

The first aim of this investigation—to provide a framework for the design and construction of web dashboards based on live data streams—has been successfully addressed. Implementation proved simple and fast. The management of interactions, both client visualizations/controls → server and server → clients visualizations/controls, was a natural consequence of state sharing. Advanced real-time features like a chat log, confirmation messages, and record notifications were directly implemented. The second objective—formulating a user-centric design methodology—pointed out the necessity of a mouse-pointer tracking system that logs each user's visits.

The creation of such a tracker is considered the third goal for this line of research. The elaborate mouse-pointer visitation log can be exploited to enable advanced user-centric features, including the sharing of mouse pointers while visiting a web page, mouse pointers being purposefully driven by a third party to highlight areas of interest, or the recording and replay of a visitor's session. These features will enhance the user experience and constitute the official phases five and six.

Fig 10 . 4 : Research Progression: Achieved Aims & Future Phases

10.8.1. Summary and Final Thoughts

It is shown that the fundamental concept of dashboards and the purpose it has in the business world is to display useful and relevant information, enabling human and automated decision-making, and control. A web dashboard has been implemented, displaying real-time information produced by a live streaming Twitter application, with a world map focused on the discussion about the COVID-19 pandemic, along with a time series. Additional controls allow the users to specify their language and country of interest for a customized informative visualization. The web dashboard has been developed using the Shiny framework, allowing the creation of reactive dashboards with a more user-friendly programming approach.

A qualitative comparative analysis and evaluation were carried out, addressing the effect that the use of frameworks has on the dashboard development process. The main aspect highlighted by the participants in the evaluation was the significant reduction in the development needed when dashboards are built, providing support for reactive programming. These observations are in line with the authors' conclusions. Another point not included in the comparative analysis and evaluation is related to the internal organization of the code. When Shiny's reactive programming is exploited, it is possible to organize the code in a simple and concise way, improving its maintainability.

References

- [1] Duarte A. (2023). User-Centric Data Visualization: Creating Dashboards Through Human-Centered Design. IGI Global. <https://doi.org/10.4018/978-1-6684-6786-2.ch010>
- [2] Pandiri, L., & Singireddy, S. (2023). AI and ML Applications in Dynamic Pricing for Auto and Property Insurance Markets. *Journal for ReAttach Therapy and Developmental Diversities*. [https://doi.org/10.53555/jrtdd.v6i10s\(2\).3611](https://doi.org/10.53555/jrtdd.v6i10s(2).3611)
- [3] Revano T F, Garcia M B. (2021). Designing Human-Centered Learning Analytics Dashboards Using a Participatory Design Approach. *IEEE HNICEM*. <https://doi.org/10.1109/HNICEM54116.2021.9731917>
- [4] End-to-End Traceability and Defect Prediction in Automotive Production Using Blockchain and Machine Learning. (2022). *International Journal of Engineering and Computer Science*, 11(12), 25711-25732. <https://doi.org/10.18535/ijecs.v11i12.4746>
- [5] Ansari B, Martin E G. (2023). Integrating Human-Centered Design in Public Health Data Dashboards: Lessons from New York State. *Journal of the American Medical Informatics Association*, 31(2):298–305. <https://doi.org/10.1093/jamia/ocad102>
- [6] Sheelam, G. K. (2024). AI-Driven Spectrum Management: Using Machine Learning and Agentic Intelligence for Dynamic Wireless Optimization. *European Advanced Journal for Emerging Technologies (EAJET)*-p-ISSN 3050-9734 en e-ISSN 3050-9742, 2(1).
- [7] Nimbarte A D, Smith N, Gopalakrishnan B. (2024). Human Factors Evaluation of Energy Visualization Dashboards. *Journal of Control, Automation and Electrical Systems*. <https://doi.org/10.1177/10648046211028693>

- [8] Raviteja Meda. (2024). Agentic AI in Multi-Tiered Paint Supply Chains: A Case Study on Efficiency and Responsiveness . Journal of Computational Analysis and Applications (JoCAAA), 33(08), 3994–4015.
- [10] Few S. (2006). Information Dashboard Design: Displaying Data for At-a-Glance Monitoring. Analytics Press. <https://doi.org/10.1002/9781119055259>