Best Practices in CubeSat Control Software Development: A Case Study of the SAGE Mission

Andrin Benz
University of St. Gallen, Switzerland

Sebastian Oes
University of St. Gallen, Switzerland

Johannes Schöning
University of St. Gallen, Switzerland

Abstract

The increasing complexity of CubeSat missions necessitates mission control software that is both efficient and user-friendly. This paper describes the design and implementation of a multi-mission CubeSat user interface for the SAGE Mission, a student-led CubeSat program. The system includes a web-based interface that provides telemetry visualisation, automated job scheduling, and real-time monitoring. Its architecture is designed to be modular, scalable, and accessible, allowing integration with multiple ground stations and support for various mission configurations. Input from stakeholders played a crucial role in shaping the interface through user evaluations, expert feedback, and digital twin simulations. Our findings highlight the significance of user-centred design in space mission software, particularly in educational and resource-constrained settings. The CubeSat SAGE mission control software enhances the accessibility of multi-mission operations and offers valuable insights into the future of space system interfaces.

2012 ACM Subject Classification Human-centered computing → Interactive systems and tools

Keywords and phrases Automated mission operations, CubeSat operations, Ground segment software, Mission control software (MCS), Multi-mission operations, Satellite mission planning, SpaceCHI Telemetry data visualisation, User interface design

Digital Object Identifier 10.4230/OASIcs.SpaceCHI.2025.2

1 Introduction & Motivation

CubeSats have revolutionised space access through their compact size and cost-effectiveness, enabling broader participation in space exploration [18]. However, managing multiple CubeSat missions introduces coordination and resource management challenges [9]. An efficient mission control software (MCS) with a well-designed user interface (UI) is essential for mission success.

Effective mission control software must provide robust technical functionality and ensure usability, trust, and efficiency in high-stakes environments. The SAGE mission, a student-led CubeSat mission, incorporates those design principles that enhance operator interaction, automate routine tasks, and mitigate cognitive load, ultimately improving mission efficiency and reliability.

This paper presents the design and implementation of a multi-mission CubeSat control software for the SAGE mission. The project addresses challenges in student-led missions, including high team turnover and limited experience compared to commercial operations.

SAGE is a 3-Unit (3U) CubeSat developed by ARIS. CubeSats are nanosatellites with basic units of $10 \times 10 \times 10$ cm, serving as low-cost alternatives to larger spacecraft [18]. One unit houses a biological payload, while the remaining two accommodate spacecraft systems. The mission objectives include providing an educational satellite platform in space, studying human cell ageing in microgravity, operating an amateur radio and custom GNSS equipment,

Figure 1 The multi-mission CubeSat control software for the SAGE mission in use.

and testing a custom RISC-V ASIC developed at the Integrated Systems Laboratory at ETH Zurich. The mission will be launched into a sun-synchronous Low Earth Orbit by a U.S.-based launch provider in 2026. The following teams are contributing to mission success and were involved in the design of the SAGE MCS:

- **Thermal Control and Structures (TC-STR):** Ensures the CubeSat can withstand launch and space conditions, provides mounting for other subsystems, and maintains satellite temperature within operating limits.
- **Biological Payload (PAY):** Investigates human cell ageing by hosting the first human cell culture in space.
- **Electrical Power System (EPS):** Generates, stores, and provides power using solar panels, a battery pack, and a power distribution module.
- **Communication System (COM):** Enables data reception, command transmission, and health status monitoring.
- **Attitude Determination and Control System (ADCS):** Assesses, monitors, and adjusts the satellite's attitude using sensors, actuators, and state estimation software.
- **On-Board Computer (OBC):** Manages and processes tasks across subsystems, ensures satellite health, and provides local data storage.
- **Mission Operations (MOPS):** Operates the satellite from a Mission Operations Center in Zurich, including orbit determination, health monitoring, conducting experiments, and addressing anomalies.

CubeSat operations are divided into data retrieval and job scheduling, ensuring mission objectives are met. Data retrieval involves acquiring telemetry and payload data from the satellite. Raw data packets are received via ground station antennas, decoded, processed, and stored in a central database for analysis. Data integrity and redundancy are crucial for anomaly detection and mission performance assessment.

Job scheduling pertains to defining, verifying, and executing commands to control satellite operations. Users define jobs via a UI, which is then verified using a digital twin simulation. Verified jobs are uplinked to the CubeSat and executed at the scheduled time, with each step logged for traceability and troubleshooting.

2 Related Work

The development and maintenance of operations centres for satellite missions have been the subject of past research. Traditional satellite operations centres are often designed for single missions and may not provide an efficient foundation for managing multiple concurrent (CubeSat) satellite missions. This has led to a shift towards the adoption of multi-mission software solutions in recent years.

To support the simultaneous management of multiple satellite missions, the concept of a Multi-Mission Operations Centre (MMOC) has been introduced. In collaboration with European national space agencies, the European Space Agency (ESA) developed the European Ground Systems Common Core (EGS-CC)[1], providing a shared infrastructure to facilitate the implementation of different missions [15, 2]. Similarly, NASA's Multi-Mission Operations Center (MMOC)[2] and the Advanced Multi-Mission Operations Service (AMMOS)[3] offer frameworks that enable the efficient operation of multiple missions using shared resources [9].

CubeSat missions present special operational requirements due to their inherent resource limitations and relatively short mission lifespans [10]. Traditionally, space missions have focused on the development of high-cost, large-scale satellites, but there has been a growing trend towards smaller satellite systems. CubeSats, particularly, have gained popularity among educational institutions, space agencies, and industry stakeholders due to their versatility and rapid development cycles [1].

At SpaceCHI, although not the core focus, related aspects of crew autonomy and space system development have been discussed in previous editions of the conference. Already in 2007, Huang et al. [5] explored the use of large interactive displays for workgroup collaboration, noting that while many systems have been tested in labs, they face challenges when deployed in real-world settings. Their study examines the adoption of NASA MERBoards during the Mars Exploration Rover missions, identifying deployment challenges external to the system design, and discusses how these issues apply to the broader use of shared ubiquitous computing technologies. More recently, Zheng et al. [21] emphasise the growing need for crew autonomy during long-duration missions, particularly in mission planning, due to the communication delays inherent in deep-space exploration. They propose a mixed-initiative approach where a computer assists in managing complex tasks like constraint handling while still allowing astronauts to retain control. The Playbook tool, designed for this purpose, addresses these challenges with its innovative scheduling capabilities, supporting autonomous crew planning.

[1] http://www.egscc.esa.int/
[2] https://www.nasa.gov/mmoc/
[3] https://ammos.nasa.gov/

Similarly, Nilsson et al. [13] critique traditional analogue environments for space system development, highlighting their high costs and limited participation. To counter these limitations, they are developing a mixed-reality testbed that combines physical mockups with digital extensions. This more accessible and cost-effective approach aims to facilitate participatory design and rapid prototyping, ensuring that the diverse future users of space systems can contribute to their development.

Building on these concepts of autonomy and more flexible and modular system design, Karasinski [6] stresses the necessity of enhanced crew autonomy in future space missions due to significant transmission delays and reduced bandwidth. They advocate for technologies that improve situational awareness (SA) for both astronauts and remote mission control. Their paper outlines a usability study on integrated timeline tools, which aim to predict and resolve potential mission constraints, ultimately supporting more successful and autonomous space missions.

Heinicke et al. [4] advocate for an even broader perspective that views inclusion not merely as a gesture of diversity but as a means to improve mission outcomes. They propose that space agencies should integrate individuals with various disabilities into the design and planning processes of space missions, thereby fostering innovation and resilience in space exploration.

While distinct in their approaches, these contributions underscore the importance of integrating autonomy, modular design, and innovative design methods to address the unique challenges of future space missions as we aim to reflect with our new UI for the CubeSat mission.

The design of UIs plays a crucial role in satellite mission operations, enhancing operator efficiency and minimising errors. Effective UI design requires the involvement of domain experts in the decision-making process [19]. NASA's Open Mission Control Technologies (OpenMCT)[4] is an open-source initiative aimed at providing customisable and extensible mission control UIs [3].

Dashboard design and data visualisation are critical for monitoring satellite operations effectively. Well-designed dashboards consolidate data from various sources, offering a comprehensive view of multiple mission statuses and improving decision-making and historical analysis [7]. The choice of visualisation techniques significantly influences the clarity and interpretability of complex data, ensuring that operators can extract meaningful insights [14]. Research highlights the importance of real-time data visualisation in operational contexts, aiding in the rapid analysis of evolving scenarios [17].

A crucial aspect of dashboard design is the integration of data from multiple sources to enhance situational awareness and historical analysis [7]. Selecting appropriate data visualisation methods ensures that complex information is presented in an accessible manner, allowing operators to identify critical insights swiftly [14]. User-centric design principles further enhance dashboard usability by prioritising intuitive layouts and ensuring the timely accessibility of relevant data [11]. Engaging stakeholders throughout the design process helps tailor dashboard features to specific operational requirements, leading to improved user adoption [20].

To maximise effectiveness, dashboards should incorporate key performance indicators (KPIs) relevant to mission operations, facilitating real-time monitoring and control [14]. The inclusion of feedback mechanisms allows continuous improvement based on user experiences

[4] `https://nasa.github.io/openmct/`

Figure 2 Example Dashboard for the multi-mission CubeSat control software for the SAGE mission.

and operational outcomes [8]. However, challenges such as data overload and the need for ongoing operator training must be considered to optimise dashboard utility and ensure long-term usability.

Despite advancements in multi-mission control software, CubeSat operations, and UI design, existing solutions often lack the simplicity and accessibility required for effective multi-mission CubeSat operations centres. Many systems do not provide integrated, user-friendly dashboards that balance flexibility, customisation, and intuitive design. Non-configurable mission control software that is tightly coupled to a specific mission or mission type leads to additional development overhead that does not align with the rapid development cycles of CubeSat missions. This research aims to address this gap by developing dashboards that are easy to implement and operate, guided by user-centred design principles and informed by operator feedback through user interviews.

3 User Interface Design

Therefore, designing an effective UI for mission control software requires balancing usability, cognitive load reduction, and adaptability to diverse operational roles. As stated in the previous section, related research highlights the importance of usability in such interfaces [12] to facilitate intuitive interaction. Building on the related work, the UI of the SAGE MCS was developed to ensure efficient data visualisation, seamless navigation, and adaptability to mission needs. This section outlines the UI architecture, key design principles, and findings from our iterative evaluation process.

3.1 Concept and Modular Design

The SAGE MCS UI is structured into three primary modules: theme, page structure, and views. This modular approach enables parallel development, ensuring that modifications to one component do not impact others. The initial design considerations focused on visual

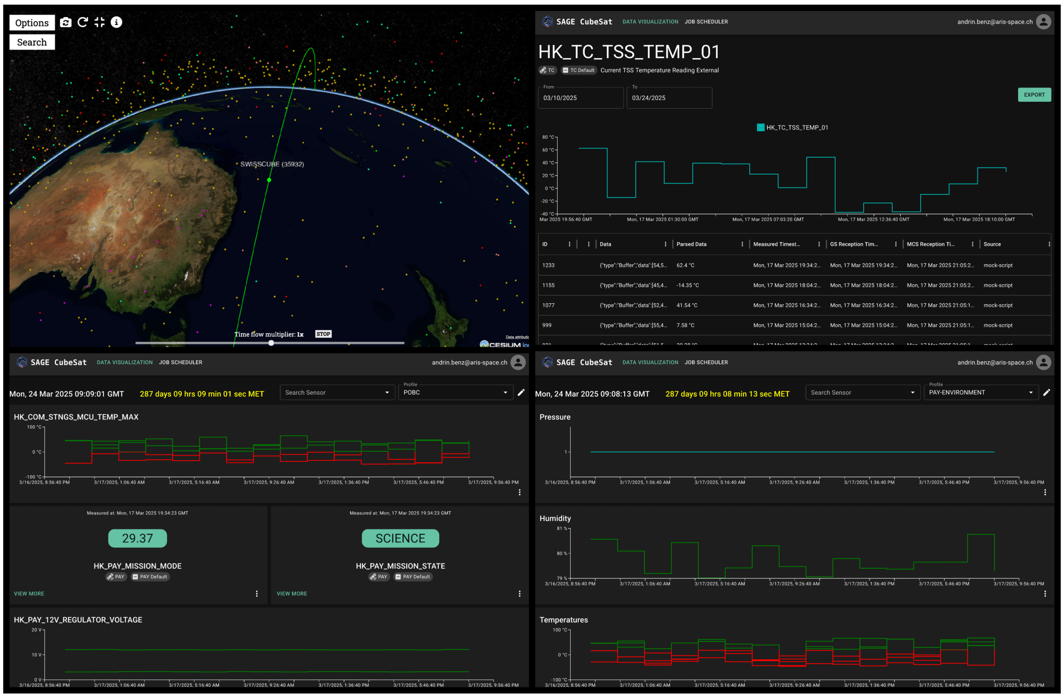

Figure 3 Alternative Dashboard Configurations for the multi-mission CubeSat control software for the SAGE mission.

clarity, consistency, and responsiveness to support operators in high-stress mission scenarios. Key principles included using visual cues to distinguish interactive elements, maintaining consistency in icons, colours, and layout to facilitate learning and reduce errors, and providing immediate feedback through hover effects and visual confirmations to enhance user confidence. Additionally, minimising cognitive load was a priority to enable users to concentrate on mission-critical tasks rather than interface navigation.

The SAGE MCS UI is structured around a modular dashboard layout that enables users to tailor their views based on specific operational roles. This approach ensures that mission-critical data is presented in a clear and accessible manner while allowing for flexibility in adapting to different user needs. Figure 2 shows an example of the dashboard used in the multi-mission CubeSat control software. Figure 3 shows alternative configurations of the dashboard.

The dashboard incorporates several key controls essential for mission operations. A Zulu Time display provides real-time system time for synchronisation across mission teams. Alongside this, a Mission Time counter tracks the elapsed duration of the mission, ensuring precise timing for operational procedures. To facilitate rapid access to telemetry data, the dashboard includes a Sensor Search function, enabling users to locate specific sensor readings efficiently. Additionally, the Profile Selection feature allows users to switch between predefined or customised dashboard profiles, ensuring that the displayed information aligns with their specific tasks and responsibilities.

The system supports configurable dashboards that can be dynamically adjusted to match operational requirements. Each dashboard is based on modular cards, which serve as the primary means of visualising mission data. Users can select from predefined profiles or create custom configurations that reflect their individual workflows. This flexibility ensures that mission operators can focus on the most relevant information without being overwhelmed by extraneous data.

A diverse set of cards is available within the dashboard, each designed to support different types of data representation. Static cards provide essential reference materials, including textual descriptions, images, and quick-access links to external resources. Data cards offer a more interactive means of presenting mission telemetry, utilising charts, tables, maps, and real-time visualisations to display evolving sensor data. In addition, mission-specific cards cater to the unique requirements of space operations, featuring tools such as 3D satellite models, actuator state visualisations, system alerts, and sensor imagery. These elements collectively contribute to a comprehensive and adaptable interface supporting routine monitoring and critical decision-making.

To ensure consistency, usability, and accessibility across various devices, the UI adheres to Material Design principles [16]. This design framework provides a structured approach to visual hierarchy, interaction patterns, and responsive layouts, allowing for seamless operation across different screen sizes and platforms.

4 Evaluation

An iterative evaluation process was conducted to ensure that the design met operational requirements, involving key stakeholders throughout the development cycle. This approach ensured continuous feedback, enabling the refinement of system functionalities to better support mission operations.

4.1 Initial Stakeholder Discussions

The development process commenced with structured meetings involving representatives from the Mission Operations (MOPS), Payload (PAY), and On-Board Computer (OBC) teams. These discussions provided an initial overview of the system's objectives and constraints. Following a collaborative session, the fundamental architectural decisions were established: the software would be web-based, hosted externally, and structured around a REST API to ensure scalability and interoperability. This architecture was selected to support seamless data exchange between subsystems while maintaining a user-friendly interface for mission operators.

4.2 User Story Development

Each subteam was tasked with defining user stories to ensure that the system aligned with mission needs. These user stories captured the specific requirements of operators and engineers, detailing the interactions necessary to monitor and control mission parameters effectively. The user stories were compiled into a structured document, serving as a reference for feature prioritisation. The document queries users for specific information related to each interaction with the system. This provides the development team with a clear understanding of which system components are involved in a given interaction, which parameters are passed, and what return values and post-conditions are expected. On top of functional requirements, the user stories provided non-functional stakeholder requirements. One of the key insights that emerged from this process was the necessity for flexible, role-specific dashboard profiles. Given the distinct responsibilities of different operators, it became evident that a configurable interface was essential to prevent information overload and streamline workflows.

4.3 Iterative Development and Feedback

Following the initial requirement gathering, an iterative development cycle was established. Weekly review sessions were held with stakeholders, allowing for continuous refinement of the UI and dashboard configurations. These iterative improvements were informed by direct user feedback, ensuring that practical operational needs guided design choices.

Once the core dashboard functionality was implemented, a usability workshop was conducted with representatives from each subteam. Participants were provided with early interface prototypes and tasked with evaluating their effectiveness in supporting their workflows. They completed structured questionnaires, providing insights into usability, clarity, and feature relevance. The feedback gathered from this session directly influenced the refinement of dashboard layouts, interaction patterns, and data visualisation methods.

4.4 Expert Evaluation at GSOC

To validate the system in a real-world operational context, an expert evaluation was conducted at the German Space Operations Center (GSOC) of the German Aerospace Center (DLR). During this visit, the development team had the opportunity to observe mission control rooms used for the operation of satellites such as TerraSAR-X and TanDEM-X. These observations provided valuable insights into best practices for mission-critical interfaces, particularly in terms of information presentation, workflow integration, and system responsiveness.

A structured feedback session was held with domain experts in mission planning, communications, and operations engineering. This session confirmed several key design assumptions and highlighted critical aspects of interface usability. One of the primary findings was the importance of data abstraction: users should only be presented with information relevant to their specific tasks. For instance, a payload operator requires access to subsystem-specific data, while a communications engineer focuses on signal acquisition metrics. Presenting excessive or irrelevant data can lead to cognitive overload, reducing situational awareness and increasing the likelihood of errors.

The SAGE MCS dashboard's role-based configuration aligns with these principles, allowing users to customise their views according to their operational needs. By filtering information based on user roles, the system ensures that mission-critical data is presented in a clear and accessible manner, optimising decision-making and operational efficiency. The expert evaluation at GSOC validated this approach, reinforcing the importance of configurable dashboards in mission control environments.

5 Implementation & Operation

Prior to developing the SAGE MCS, a collaborative process involving all relevant stakeholders was done to formulate the technical requirements, ensuring alignment with mission objectives. Three core principles guided the infrastructure design: prioritising essential features for mission success, ease of installation and learning through automation, and resilience to ensure continuous operation. These principles shaped a robust and efficient infrastructure, optimised for functionality and durability.

5.1 Hosting and System Architecture

The SAGE MCS is hosted externally on cloud servers, selected based on key requirements such as high availability, log retrieval, rollback capabilities, manageable costs, and strong security measures. This ensures uninterrupted access, system integrity, and compliance with

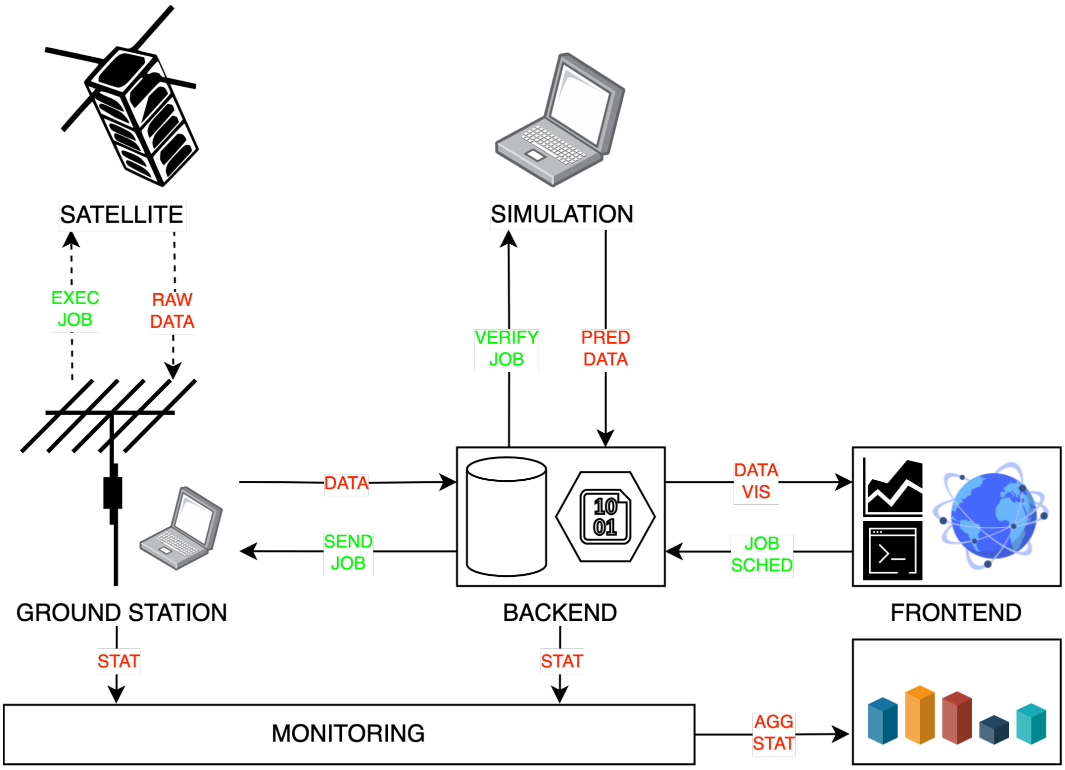

Figure 4 Overview on the setup.

industry standards, supporting the Mission Operations Centre. The MCS features a resilient core component that interfaces with external services through API calls, ensuring scalability and availability across CubeSat missions, as shown in Figure 4.

5.2 Backend, Frontend, and Simulation Integration

The backend, built with Node.js, TypeScript, and Express.js, follows the Router-Controller-Service (RCS) pattern to enhance scalability with robust error handling and logging. The Drizzle ORM ensures efficient data management. The UI, developed using React and Next.js, provides a responsive, modular interface with enhanced visualisation and interaction through React Material UI, Three.js, and Leaflet. The system also integrates with a simulation module for real-time satellite state modelling, ensuring command sequence verification in a risk-free environment. Additionally, the UI supports multiple ground stations for redundancy, automatically handling duplicate sensor data and ensuring broad mission compatibility.

5.3 Monitoring, Alerting, and Communication

The monitoring system, powered by Grafana and Prometheus, tracks performance and health metrics, generating alerts for any anomalies. Iris incident response software ensures quick resolution of issues by automatically notifying engineers via Slack. Communication between services is managed through secure RESTful APIs with error handling and logging to maintain system integrity.

5.4 Quality Assurance and Documentation

The system uses CI/CD pipelines, linting, and automated testing to ensure high-quality code, with unit and end-to-end tests validating system functionality. Mock data generation scripts can support stakeholder feedback by simulating satellite states and system behaviours. Comprehensive documentation, using Docusaurus, supports knowledge transfer and system maintenance, ensuring the sustainability of the mission control software across multiple team members that regularly change in student-led missions as well.

6 Conclusion

The SAGE MCS demonstrates a modular and flexible approach to multi-mission CubeSat control software. By prioritising simplicity, user-centric design, and comprehensive documentation, the system effectively addresses key challenges faced by student-led CubeSat missions, including high team turnover and constrained resources.

The iterative development process, guided by continuous stakeholder feedback, has resulted in a robust architecture and intuitive UI that aligns with the operational requirements of the SAGE CubeSat mission. This ensures reliable mission operations while enhancing operators' situational awareness and decision-making.

Future work could explore extending to support a broader range of missions and vehicles, including robotic systems for terrestrial and space applications. The architecture was designed with scalability in mind, facilitating such expansion.

References

1 M. V. Fernandez. Mission analysis of qb50, a nanosatellite intended to study the lower thermosphere, 2011. Bachelor thesis, University of Liège, Liège, Belgium.

2 M. Götzelmann, L. Tucker, J. Sanmartí, and N. Mecredy. The design of the european ground systems – common core (egs-cc). In *Proceedings of the SpaceOps Conference*, Pasadena, CA, USA, 2014.

3 C. Hacskaylo and T. Truong. Open mission control technologies (openmct), 2021. [Online]. Available: `https://nasa.github.io/openmct/`.

4 Christiane Heinicke, Marcin Kaczmarzyk, Benjamin Tannert, Aleksander Wasniowski, Malgorzata Perycz, and Johannes Schöning. Disability in space: Aim high. *Science*, 372(6548):1271–1272, 2021.

5 E. M. Huang, E. D. Mynatt, and J. P. Trimble. When design just isn't enough: The unanticipated challenges of the real world for large collaborative displays. *Personal and Ubiquitous Computing*, 11:537–547, 2007. `doi:10.1007/S00779-006-0114-3`.

6 J. A. Karasinski, J. Zheng, M. Yashar, and J. J. Marquez. Integrating mission timelines and procedures to enhance situational awareness in human spaceflight operations. In *SpaceCHI 2.0 Workshop*, 2022.

7 H. Khalil and I. Ben-Israel. A unified situational awareness dashboard for spacecraft operations: An integrated, fleet-centric, cloud computing based solution. In *Proceedings of the IEEE Aerospace Conference*, 2014.

8 A. Khan. Dashboard design, 2024. `doi:10.1007/979-8-8688-0119-8_5`.

9 A. Y. Ko, P. F. Maldague, T. Bui, D. T. Lam, and J. C. McKinney. The evolvable advanced multi-mission operations system (ammos): Making systems interoperable. In *Proceedings of the SpaceOps Conference*, Huntsville, AL, USA, 2010.

10 J. R. Kopacz, R. Herschitz, and J. Roney. Small satellites: An overview and assessment. *Acta Astronautica*, 2020.

11 M. Madden, E. Cary Jr., T. Esposito, J. Parker, and D. Bradley. Lessons learned from engineering a multi-mission satellite operations center. In *Proceedings of the AIAA SPACE Conference & Exposition*, Long Beach, California, 2005.

12 J. Nielsen. 10 usability heuristics for user interface design, 1994. URL: `https://www.nngroup.com/articles/ten-usability-heuristics/`.

13 T. Nilsson, L. Bensch, F. Dufresne, F. Rometsch, P. de Medeiros, E. Guerra, F. Saling, A. E. Casini, and A. Cowley. Out of this world design: Bridging the gap between space systems engineering and participatory design practices. In *Proceedings of SpaceCHI 3.0, Conference on Human-Computer Interaction for Space Exploration*, MIT Media Lab, MA, USA, 2023.

14 D. Orlovskyi, A. Kopp, and V. Y. Kondratiev. Development of a model and a software solution to support the analytical dashboards design problem. *Modern Problems of Radio Engineering, Telecommunications and Computer Science*, 1, 2020.

15 M. Pecchioli and J. M. Carranza. Highlights of the european ground system – common core initiative. In *Proceedings of the ICALEPCS Conference*, Barcelona, Spain, 2017.

16 A. Pinandito, H. M. Az-zahra, L. Fanani, and A. V. Putri. Analysis of web content delivery effectiveness and efficiency in responsive web design using material design guidelines and user centered design. In *Proceedings of the International Conference on Sustainable Information Engineering and Technology (SIET)*, Malang, Indonesia, November 2017.

17 J. J. Thomas and K. A. Cook. A visual analytics agenda. *IEEE Computer Graphics and Applications*, 2006.

18 A. Toorian, K. Diaz, and S. Lee. The cubesat approach to space access. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, USA, 2008.

19 J. Trimble and C. Webster. Agile development methods for space operations, 2012. URL: `https://www.researchgate.net/publication/269054704_Agile_Development_Methods_for_Space_Operations`.

20 A. Widera, B. Hellingrath, and C. Bubbich. Humanitarian logistics dashboards design-related requirements analysis. In *Proceedings of the IEEE Global Humanitarian Technology Conference (GHTC)*, 2015.

21 J. Zheng, S. M. Shelat, and J. J. Marquez. Facilitating crew-computer collaboration during mixed-initiative space mission planning. In *SpaceCHI 3.0 Conference on Human-Computer Interaction for Space Exploration*, 2023.