Human-Computer Interaction & Emerging Technologies, Vol. 233, 2026, 487–493
https://doi.org/10.54941/ahfe1007548

Designing Experiment Software Optimized for Data Yield, Immersion, and Control in Naturalistic Human Factors Experiments

Cara Spencer[1,2], Spencer Anderson[2], and Leanne Hirshfield[1,2]

[1]Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA
[2]Institute of Cognitive Science, University of Colorado Boulder, Boulder, CO 80309, USA

ABSTRACT

We present the Mars Investigation and Navigation Dashboard (MIND), an Unreal Engine–based platform for building and executing configurable, high-fidelity experiments. The MIND was developed to answer the call for experiment software to elicit more natural responses in human factors studies. Its design prioritized ecological validity, usability, and documentation. The participant interface was inspired by real operations software and provides an immersive 3D experience. The system also offers a modern interface that lets experimenters adjust parameters without modifying source code. For downstream analysis, the MIND generates comprehensive internal logs and supports external synchronization via Lab Streaming Layer and outputs to Robot Operating Systems. We illustrate how configurable orchestration, rich logging, and user-centered interfaces can reduce research iteration costs and expand the design space for more naturalistic studies.

Keywords: Experiment testbed, Human-computer interaction, Ecological validity

INTRODUCTION

Naturalistic studies are an integral part in investigating authentic, real-world behaviors but these insights can come at the cost of data yield and clear causal inferences. Thus, researchers across domains are moving to blend this paradigm with experimental methods by enhancing ecological validity of lab experiments where experimental controls and other modalities are more easily implemented (Nati et al., 2017). Increasingly, experiments include realistic interfaces, open ended or self paced trials, multimodal sensing, and interactive components that must remain replicable across participants (Rogers, 2011). These features are especially important as the digital world becomes more elaborate and distinct from predecessor graphics. Human factors and Human-Computer Interaction researchers have been at the forefront of much of this work but it behooves us to continue the effort and push the frontiers of experiment design and software. In practice, however, research teams often face a tradeoff: high-fidelity interactive environments can be expensive, complicated to modify or difficult to connect to other platforms. On the other hand, lighter-weight tools may not support dynamic interactions or high fidelity elements like

Received March 18, 2026; Revised April 26, 2026; Accepted May 10, 2026; Available online July 20, 2026

© 2026 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

dynamic interfaces or 3D environments. These constraints slow iteration, risk diminishing ecological validity and control, and limit interconnectivity with other platforms (Cooke et al., 1999).

Figure 1: Research environments where the MIND software has been deployed. a) Depicts a traditional lab environment with a monitor, keyboard and mouse. b) Depicts a Mars rover simulator with surround TV screens, laptops, and a joystick. c) Depicts the work stations at the Mars Desert Research Station in Hanksville, Utah. d) Depicts the fMRI set up with a participant in the control room and the other participant in the scanner.

To address these gaps, we built the Mars Investigation and Navigation Dashboard (MIND), an Unreal Engine–based experiment platform designed around three priorities: (1) usability: enabling experimenters without programming experience to configure sessions rapidly and easily connect to a variety of software and hardware, (2) documentation: capturing an end-to-end record of each session via comprehensive internal event logging and optional Lab Streaming Layer (LSL) streaming for alignment with external sensors, and (3) ecological validity: providing participants with a high-fidelity, frictionless interface in various research environments. Notably, the MIND was deployed in several research environments including a traditional lab setting with a monitor, keyboard and mouse; a rover simulator with several large TV screens, a laptop, and joystick; a field location called the Mars Desert Research Station using laptops only; and during functional magnetic resonance imaging (fMRI) using the scanner screen, laptops and joysticks (each pictured above in Fig. 1). For future versions of our studies, the system also supports an RPC-based messaging system and connecting to an external AI agent to enable adaptive or interactive components using the Robot Operating System (ROS). These priorities and features have reduced the amount of time spent piloting, training staff to run the software, and organizing data from our multiple modalities after data collection. Moreover, they allow us to reuse the testbed for different experimental paradigms rather than building additional software further reducing cost and time.

DESIGNING THE SYSTEM

Scientifically, the MIND supports investigating the mental states and behaviors of teams doing a series of investigations "on Mars." Our design draws inspiration from areas where human factors researchers have played a key role: planning and command-and-control interfaces that support rapid situational assessment and structured workflows; research in the wild paradigms; and experiment-builder tools that prioritize reproducibility through parameterization, scripting, and standardized logging (Balaban et al., 2006; Torres and Grethen, 2002; Balestrini et al., 2020; Rogers and Marshall, 2017; Wilson and Tewdwr-Jones, 2021). Before building our platform, we investigated planning tools in use at spaceflight and military command centers and cognitive task analyses done on planning and executing operations (Miller et al., 2017; Oryschak, 2020; Volz and Dorneich, 2020). We also explored the recent influx of online experiment building platforms that have accelerated experiment testbed development (see (Research Navigator, 2024) for a recent list). Many of the latest platforms provide support for designing and collecting highly controlled experiments. To complete our assessment of the state of the art, we examined 3D simulators used for experiments (excluding virtual reality since that was not our focus) which ranged from desktop-based 3D to multi-screen mock-ups (Del Grosso and Sirota, 2019; Sudár and Csapó, 2024; Li et al., 2016; van Weelden et al., 2022). Across platform types, we found needed advancements in technical and scientific capabilities. Yet, there were still some limitations to what is available to use, including limited dyadic or group-based participation, logging, customization, fidelity, and complexity. Thus, the MIND was designed using a multi-step user-centered process that integrated considerations from psychology, human factors and ergonomics, and experimental software engineering. Our design process emphasized the needs of three stakeholder groups: (1) participants, who require a usable, accessible, and immersive task interface; (2) experimenters, who need to design and perform several controlled but realistic experiments without major code changes every study; and (3) analysts, who require comprehensive, synchronized, and well-structured data outputs for downstream behavioral and multimodal analysis. We performed iterative prototyping over the course of nine pilot studies where participants used the system and could provide real-time feedback. Their performance and choices were noted by the development team and the participants were interviewed about their experience afterward. Our diverse sample of pilot participants (n = 18, age range: 18–43, 50% female) provided needed insight about accessibility, usability, and fidelity since the naturalistic nature of the task elicited a variety of interactions. Each pilot used increasingly refined versions of the software to align the platform with our requirements while maintaining flexibility across study contexts.

SYSTEM OVERVIEW

The general experiment flow supported by the system is modeled after naturalistic investigative operations. For a given scenario, there is (1) a planning phase where participants get information regarding a Mars mission, (2) an execution phase where they carry out their mission in a rover traversing a 3D Martian environment (if completing the experiment virtually), and (3) a debrief phase where participants can interact with other entities on Mars and reflect on their mission in a survey. Within these phases are several configurable elements: phase duration, timed pop-ups for guidance or experiment content, baseline blocks (e.g., a one-minute crosshair), embedded static and dynamic content (e.g., surveys, maps, in-simulation GPS), and branching logic for conditions and operator decisions. We have prebuilt seven different scenarios; each have unique information and 3D maps based on real topography from our field research environment in Hanksville, Utah.

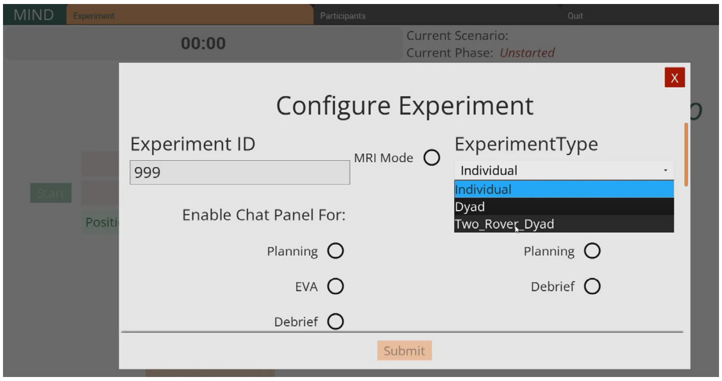

Figure 2: Depiction of the beginning of the experiment configuration selection screen. There are several experimental parameters to choose from. Those visible in this screen capture include ID, participant configuration under ExperimentType, and which phase to enable the chat panel for. Covered by the ExperimentType, is the selection for which phase the anonymous alert module is available for.

The MIND is split into server and client modes. Both adopt modern browser layouts (e.g., persistent navigation, panels, and structured views) and were informed by planning software that emphasizes visibility into task state, information sources at a glance, and clear upcoming steps. Server-side, experimenters can inspect the current phase, preview upcoming elements, and adjust configuration parameters (e.g., number of participants which is one to two, phase durations, pop-up frequency, feature access) through a centralized control interface (Fig. 2). At any point, the researcher can stop or skip the current phase or subphase, and adjust who is driving the rover. Additionally, there are several debugging tools researchers can use if they encounter a bug.

Client-side, participants view experiment content in this browser style as well to take advantage of existing familiarity with similar applications and reduce workload (Fig. 3). Participants are able to navigate the interface at their own pace and choose what to interact with. There are, however, some experimental constraints to their freedom such as preventing an individual in a dyad from moving forward without their partner or skipping a question without confirmation they mean to. Our cross-environment capabilities have some inherent control in that the interface is nearly uniform in design and the experiment flow can have identical configurations. We took care to blend the experimental constraints into the natural work flow but some interruption and differences were not avoidable—hearkening back to the previously mentioned tradeoff in high fidelity experiment design.

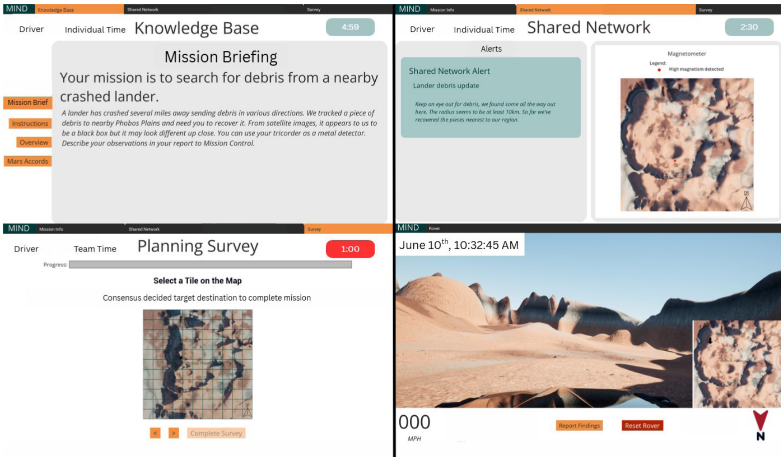

Figure 3: Depiction of the planning tabs and 3D environment from the desktop participant view. The planning tabs include the upper row and bottom left quadrants. They depict the mission briefing, mission information on what we call the Shared Network, and the final survey where participants report their planning decisions.

SYSTEM COMPONENTS

Hardware

Two to three computers are used to run MIND. One computer is used as the server and the other computer(s) are clients used by participants. The server computer must have a strong CPU to manage testbed state and handle client requests. The client computers, in addition to strong CPUs, require a dedicated graphics card to run optimally, as they're performing 3D simulation during the experiment. In addition to typical keyboard and mouse controls, USB joysticks are also supported as input devices for participants to use to drive the rover. The machines must all be on the same Local Area Network (LAN) to communicate with each other. This also implies that none of the machines can use a VPN, network proxy, or any other atypical networking.

Software

The software was developed using Unreal Engine 5.3 for the entire stack, which itself is built on top of C++. GitHub was used for source control during development. The implementation is a mixture of C++ and Unreal's "blueprint" language. Because the amount of data and assets is small, there is no database layer. Most experiment content is imported from small PNG and CSV files to maximize ease of use. Therefore, all data is compiled into the binary itself using Unreal's native data tables and asset browser. The UI is built using Unreal UI widgets. The software is compiled as a Windows desktop application which can be run on any Windows device running Windows 8+.

When client computers are connected and an experiment is started, the server operates as a state machine, orchestrating the phases of the experiment (e.g., planning, execution, debrief). The server's phase orchestration engine decides when the experiment needs to change phases, and the server will announce to the clients when phase changes happen (e.g., moving from the planning phase to the execution phase). The server is also responsible for maintaining state of the experiment (e.g., how far each participant has made it through a survey, where the rover object is in the 3D environment). This is critical for the engine to know when to progress the phase.

All activity in the testbed is output to a file on the server computer. This data includes all participant interactions with the testbed, answers to survey questions, and changes in experiment state (e.g., whenever the server changes phases). For ease of consumption, the survey questions and answers are also exported to a separate file in isolation. Certain triggers are also output to LSL and ROS to be compatible with external sensor and AI systems.

User Experience

To operate the MIND, a researcher must first connect all computers to the same LAN, either wirelessly over a private network or via ethernet and switch or router. Next, she will start the server by completing the experiment configuration using the researcher interface on the computer designated as the server. Once the server is running, the client computer(s) can connect. Devices collecting LSL or ROS inputs begin receiving data at this time if enabled. After the client computers join the server, the researcher can add metadata details to the client(s) like IDs and conditional assignments, then press start to initiate the experiment. The MIND will then automatically proceed through training scenarios if selected, a baseline for physiological sensors if selected, and the selected number of scenarios based on the configuration. Each scenario contains customizable information and sequencing. After completing the experiment, the researcher can download a complete log of all events that occurred.

CONCLUSION

We present the MIND experiment platform designed to increase ecological validity, usability, and utility. Using modular configuration parameters (e.g., number of participants, scenarios, timers, pop-ups, baseline blocks), experimenters can flexibly and simply build and execute studies without having to access the codebase—lowering the barrier to entry and increasing iteration speed. The MIND generates a complete, exported record of everything that occurs with high accuracy facilitating a spread of study methods from observation to event-based modeling to dose-effect quantification. Inspired by real spaceflight planning software and recent experiment platform developments, participants can immerse themselves into a high-fidelity experience that elicits authentic behaviors. We hope this work encourages continued discussion of how experimental platforms that prioritize ecological validity can be developed accessibly and cost-effectively.

ACKNOWLEDGMENT

This work was supported by the Air Force Office of Scientific Research (AFOSR: Award FA9550-23-1-0453).

REFERENCES

Balaban, E., Orosz, M. D., Kichkaylo, T., Goforth, A., Sweet, A. and Neches, R. (2006), Planning to explore: Using a coordinated multisource infrastructure to overcome present and future space flight planning challenges., *in* 'AAAI Spring Symposium: Distributed Plan and Schedule Management', pp. 9–16.

Balestrini, M., Gallacher, S. and Rogers, Y. (2020), Moving hci outdoors: Lessons learned from conducting research in the wild, *in* 'HCI outdoors: theory, design, methods and applications', Springer, pp. 83–98.

Cooke, N. J., Rivera, K., Shope, S. M. and Caukwell, S. (1999), 'A synthetic task environment for team cognition research', *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* 43(3), 303–308.

Del Grosso, N. A. and Sirota, A. (2019), 'Ratcave: A 3d graphics python package for cognitive psychology experiments', *Behavior research methods* 51(5), 2085–2093. Li, C.-L., Aivar, M. P., Kit, D. M., Tong, M. H. and Hayhoe, M. M. (2016), 'Memory and visual search in naturalistic 2D and 3D environments', *J. Vis.*16(8), 9.

Miller, M. J., McGuire, K. M. and Feigh, K. M. (2017), 'Decision support system requirements definition for human extravehicular activity based on cognitive work analysis', *Journal of Cognitive Engineering and Decision Making* 11(2), 136–165.

Nati, M., Gluhak, A., Domaszewicz, J., Lalis, S. and Moessner, K. (2017), 'Lessons from SmartCampus: External experimenting with user-centric internet-of-things testbed', *Wirel. Pers. Commun.* 93(3), 709–723.

Oryschak, J. E. (2020), 'Specifying space defense operator interfaces through the application of cognitive systems engineering and prototyping'.

Research Navigator (2024), 'List of useful tools & platforms for experimental psychology', https://www.theresearchnavigator.com/post/list-of-useful-tools-platforms-for-experimental-psychology. Accessed: 2025-12-16.

Rogers, Y. (2011), 'Interaction design gone wild', *Interactions* 18(4), 58–62.

Rogers. Y. and Marshall, P. (2017), *Research in the Wild*, Synthesis Lectures on Human-Centered Informatics, Morgan & Claypool.

Sudár, A. and Csapó, B. (2024), 'Comparing desktop 3D virtual reality with web 2.0 interfaces: Identifying key factors behind enhanced user capabilities', *Heliyon* 10(11), e31717.

Torres, C. M. and Grethen, M. (2002), Planning space operations-past, present & future, *in* 'SpaceOps 2002 Conference', p. 08.

van Weelden, E., Wiltshire, T. J., Alimardani, M. and Louwerse, M. M. (2022), 'Comparing presence, workload, and performance in desktop and virtual reality flight simulations', *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* 66(1), 2006–2010.

Volz, K. M. and Dorneich, M. C. (2020), 'Evaluation of cognitive skill degradation in flight planning', *Journal of Cognitive Engineering and Decision Making* 14(4), 263–287.

Wilson, A. and Tewdwr-Jones, M. (2021), *Digital participatory planning: Citizen engagement, democracy, and design*, Routledge.