Managing Multi-Agent Research Systems: A Dashboard for Human Oversight of Coordinating AI Agents

Brian Kitano
Miravoice, Silon
United States

Evan Carlson
Silon
United States

Bryan Russett
Empathic
United States

Alex Kesling
Empathic
United States

Abstract

Recent work shows that LLMs can generate research ideas rated as novel by human experts, but these evaluations stop at the proposal stage—they do not run the experiments. We argue that execution is where human oversight becomes most critical: bugs, numerical instabilities, and unexpected results require judgment that idea-level evaluation cannot capture. We report a six-week reflective deployment of a multi-agent research system used by the authors, during which the system generated and executed more than 200 proposals and experiments. Through this deployment, we observed key challenges in human oversight of multi-agent systems: agents run 24/7 and produce overwhelming information; their parallelizability amplifies the oversight burden; steering at the micro level (a single agent on a single task) differs from the need to standardize exchangeable research artifacts; and lifting the human-review bottleneck after completion requires dedicated affordances. We contribute a dashboard that addresses these challenges: synchronous and asynchronous experiment review, complete categorized traces of human/agent/infrastructure events, manual intervention points for triggering experiments and follow-up tasks, and experiment artifact interfaces. We discuss future design implications for maintaining human agency over continuously operating agents.

CCS Concepts

• **Human-centered computing** → **HCI design and evaluation methods**; *Interaction design.*

Keywords

multi-agent systems, human oversight, AI agents, dashboard design, human-in-the-loop, research automation

ACM Reference Format:
Brian Kitano, Evan Carlson, Bryan Russett, and Alex Kesling. 2026. Managing Multi-Agent Research Systems: A Dashboard for Human Oversight of Coordinating AI Agents. In *Proceedings of Human-centered Evaluation and Auditing of Language Models (HEAL@CHI'26).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

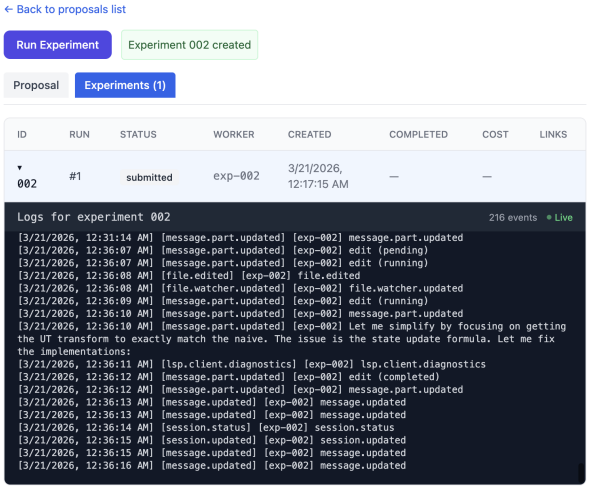

Figure 1: Experiment overview showing real-time agent status via the experiment event stream.

1 Introduction

Recent work has evaluated whether LLMs can generate novel research ideas. Si et al. [1] conducted a large-scale study with 100+ NLP researchers, finding that LLM-generated ideas were rated as more novel than human ideas, though with weaker feasibility scores. Critically, their study evaluated ideas only—they did not implement or run the proposed experiments. Yet execution is often where research goes wrong: bugs, numerical instabilities, unexpected results, and resource constraints require human judgment that idea-level evaluation cannot capture.

We extend this line of work by deploying a multi-agent system that goes beyond idea generation to *full experimental execution*, and we report lessons learned about the human oversight challenge this creates. Monitoring ongoing agent experiment progress, and evaluating their credence and fidelity to proposals after experiment completion, require higher degrees of interactivity than idea generation.

Our contribution is a set of design implications and a dashboard for managing asynchronous multi-agent systems. Drawing on a six-week reflective deployment by the authors rather than a formal user study, we identify affordances that proved necessary in practice: distinguishing infrastructure health from agent behavior, enabling manual interventions that coexist with autonomous operation, and structuring experiment results to be easily verifiable.

HEAL@CHI'26, Barcelona, Spain
2026. ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

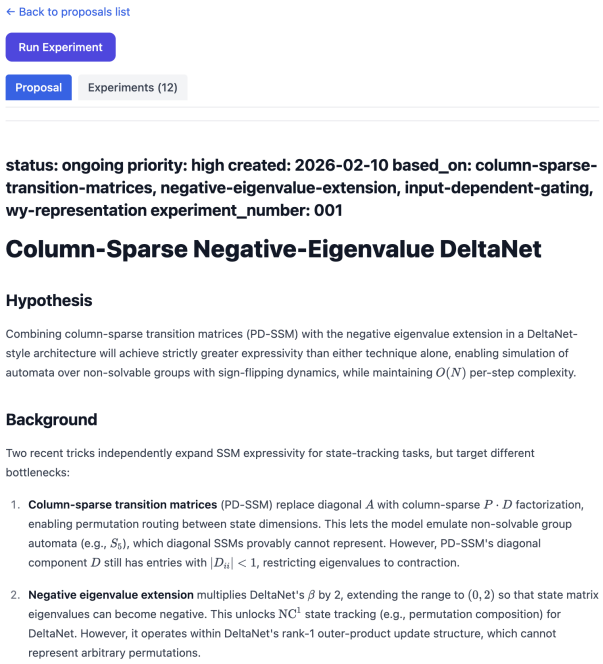

Figure 2: Proposal view, where researchers review and edit proposal details generated by the Proposal agent. Researchers can also dispatch a worker to run the experiment immediately.

2 System Architecture

Our multi-agent system targets neural architecture research, but the coordination patterns generalize to other research domains, with some caveats.

2.1 Agent Roles

- **Trick Search Agent**: Continuously parses literature and extracts computational primitives (matrix factorizations, parallelization schemes, algebraic structures). Outputs structured "trick cards" with complexity analysis and applicability conditions.
- **Proposal Agent**: Synthesizes tricks into experiment proposals. Each proposal specifies a hypothesis, mathematical formulation, minimum viable experiment (MVE) design, and success criteria. Draws on the shared trick catalog and learns from past experiment outcomes.
- **Experiment Agent**: Claims proposals, implements MVEs, runs experiments on GPU infrastructure, and reports results. Multiple experiment agents can run in parallel on different proposals.

2.2 Coordination Mechanisms

Agents coordinate through a shared event bus and shared artifacts stored in a common knowledge base that *grows continuously* as agents work. The event bus lets agents subscribe to user actions and other agents' outputs, while the shared artifacts provide durable state for review and downstream work:

- **Unified event bus**: A shared stream of events emitted by users and agents. Agents can subscribe to task-relevant events, react asynchronously, and query prior activity without requiring tightly coupled point-to-point integrations.
- **Trick catalog**: A growing repository of computational primitives, tagged by category and linked to source papers. The trick search agent adds new entries as it processes literature; over six weeks it generated 381 tricks.
- **Proposal queue**: Given fixed concurrency limits on agents and compute, experiment agents pull from a queue of experiment proposals with status tracking (pending, claimed, in-progress, completed, abandoned). The Proposal agent continuously generates new proposals; over six weeks it generated around 200 proposals.
- **Experiment logs**: Detailed records of implementation attempts, bugs encountered, and results—a learning resource that informs future agent decisions.

3 Dashboard for Human Oversight

The dashboard provides key affordances for managing asynchronous multi-agent systems.

3.1 Asynchronous Review: Timeline and Event Log

Because agents run continuously and in parallel, humans return to find hours or days of accumulated activity. A timeline surfaces what happened while the user was away: new tricks discovered, proposals generated, experiments completed or failed, and strategy updates from the log agent. Filtering by time range, agent, or event type enables rapid triage.

3.2 Infrastructure vs. Agent Monitoring

We distinguish two monitoring concerns. *Infrastructure monitoring* answers: are the agent processes running? Do they have GPU access? Are there OOM errors? (We encountered cases where experiment agents attempted training on the orchestrator machine rather than Modal workers.) *Agent monitoring* answers: are they doing the right kind of work? Are proposals drifting off-topic? The event bus separates these into separate event types because the interventions differ—restarting a crashed process versus redirecting research focus.

3.3 Manual Interventions

Autonomous operation must coexist with human initiative. Users can kick off their own experiments, bypassing the proposal queue ("queue-jumping") (Figure **??**). They can add ad-hoc tricks or proposals that agents will pick up and build on. In our deployment, these intervention points helped preserve operator agency rather than reducing the human role to post-hoc review.

3.4 Feedback Throughout the Lifecycle

Human input can enter at any point: approving or rejecting tricks before they enter the catalog, editing proposals before experiments

claim them, retrieving experiment artifacts after runs complete, and asking workers for follow-up tasks. This flexibility lets humans intervene where their judgment adds most value, which varies by context. In practice, artifact design became a core interface concern: results had to be packaged so humans could inspect code, traces, and outputs well enough to decide whether to trust or build on them.

4 Design Implications

Through iterative deployment, we identified recurring challenges and the design implications they suggest:

Design implication 1: Workflows must support asynchronous interactivity. We initially designed for real-time monitoring, but the event stream quickly becomes overwhelming. The system must assume asynchronous interaction: batch updates, asynchronous reviews of artifacts, and alerts for items requiring urgent attention.

Design implication 2: Use a unified event bus to support asynchronous coordination and context retrieval. With a unified event bus, agents can subscribe to and query events generated by users and other agents. In practice, this makes it possible to introduce task-specific agents without redesigning the whole system around bespoke integrations, and it let agents interact asynchronously while still sharing a common operational picture.

Design implication 3: Separate infrastructure monitoring from agent-behavior monitoring. Early debugging sessions revealed that many "agent errors" were actually OOM crashes, network timeouts, or jobs running on the wrong machine. Separating infrastructure monitoring from agent monitoring reduced false diagnoses.

Design implication 4: Design research artifacts to minimize verification time. It was not enough for agents to produce results; they had to produce reviewable results. Humans needed to inspect the code written by the agent, retrieve experiment artifacts after execution, and understand enough to verify that the artifacts are faithful to the proposal and don't contain errors. Designing artifacts to maximize human trust and verification therefore became a core requirement rather than a secondary presentation concern.

5 Current Status

The system was used continuously during the six-week deployment period, generating and executing more than 200 proposals and experiments. Over time, experiment-agent quality became the primary bottleneck, so we paused further proposal generation and shifted attention toward improving experiment execution and oversight. We identify the following areas of improvement:

Research artifact verification. Research artifact design deserves deeper treatment: future systems should make it easier for both humans and other agents to verify what an experiment did, what code was executed, what outputs were produced, and how those outputs should be interpreted.

Affordances during execution. User affordances during execution remain underdeveloped. Our deployment suggested that operators need better support for deciding when to interrupt, redirect, or annotate an agent while work is still in progress, rather than only before or after execution.

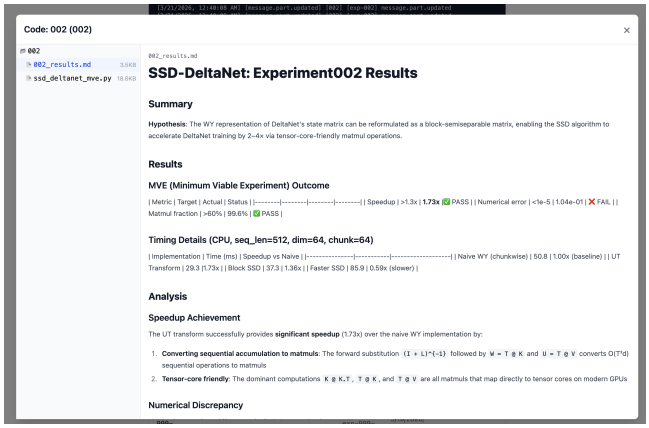

Figure 3: Results view, where an operator can review code, notes, and other artifacts generated by the agent.

Taxonomy of research artifacts. We see value in developing a taxonomy of research artifacts across scientific domains. Different sciences rely on different primary artifacts—for example code, proofs, measurements, simulations, datasets, or experimental traces—and those artifacts require different forms of verification, trust, and human oversight. A clearer taxonomy could help guide both agent design and interface design for human oversight.

Macro steering. Steering a single agent on a single task (micro) differs from adjusting overall direction and structure (macro). For micro steering, users can review and edit the code an experiment agent is about to run, or provide targeted feedback on a specific proposal (Appendix Figure 4). We believe macro-steering, or being able to command multiple agents at once, will be another important affordance to add.

6 Contributions to HEAL Themes

This work addresses HEAL's "AI Agents-in-the-Loop" theme from a systems perspective:

Async-first design: We show that multi-agent oversight must assume asynchronous human interaction, with affordances for catching up, triaging, and intervening after the fact.

Multi-level steering: We distinguish micro steering (individual agent, individual task) from macro steering (overall direction), each requiring distinct interface affordances.

Infrastructure vs. behavior monitoring: We demonstrate the importance of separating "is it running?" from "is it doing the right thing?"—a distinction that proved critical for effective debugging.

Human agency through intervention: Rather than asking whether agents can replace humans, we ask: what affordances let humans remain active participants—initiating work, jumping queues, providing feedback—alongside autonomous agents?

7 Limitations

This paper reports a reflective deployment by the authors rather than a formal user study. The observations are therefore best understood as formative evidence about interface requirements and failure modes, not as statistically grounded claims about general

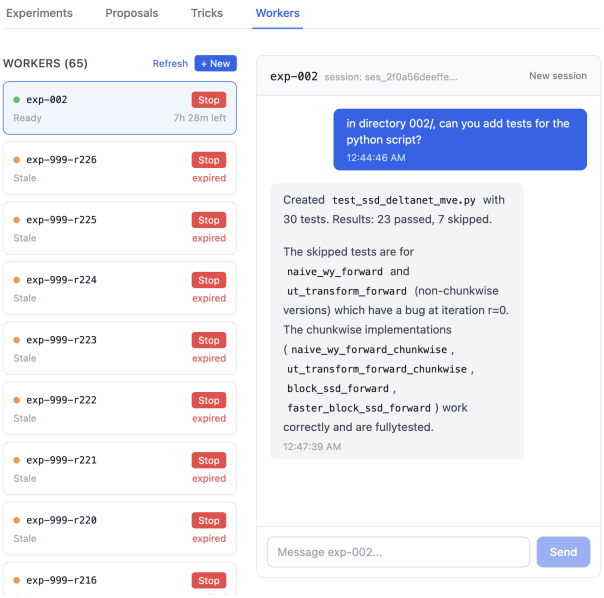

Figure 4: Micro-steering view, where an operator can ask an agent for follow-up tasks.

user preferences. Future work should validate these design implications with structured user studies and broader participant pools.

Acknowledgments

We thank Miravoice for providing compute resources.

References

[1] Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2024. Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers. *arXiv preprint arXiv:2409.04109*.

[2] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The AI Scientist: Towards Fully Automated Open-ended Scientific Discovery. *arXiv preprint arXiv:2408.06292*.

[3] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv preprint arXiv:2308.08155*.

[4] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. 2024. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. *ICLR*.

[5] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, et al. 2019. Guidelines for Human-AI Interaction. *CHI*.