

UC Berkeley

UC Berkeley Previously Published Works

Title

Design Guidelines for Building Usable Artificial Intelligence Experiences

Permalink

<https://escholarship.org/uc/item/0kq8m260>

Journal

Computing in Science & Engineering, 28(3)

ISSN

1521-9615

Authors

Cohoon, Johanna

Paine, Drew

Cohn, Marisa Leavitt

et al.

Publication Date

2026-07-01

DOI

10.1109/mcse.2026.3725613

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

Dept editors

Drew Paine, pained@lbl.gov Marisa Leavitt Cohn, marisa@itu.dk

Department: User Experience

Design Guidelines for Building Usable Artificial Intelligence Experiences

Johanna (Hannah) Cohoon, 0000-0002-3352-9766, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA

Drew Paine, 0000-0003-0711-9744, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA

1521-9615 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies.

Digital Object Identifier [10.1109/MCSE.2026.3725613](https://doi.ieeecomputersociety.org/10.1109/MCSE.2026.3725613)

Date of current version 23 September 2026.

Computing in Science & Engineering (CiSE)

Published by the IEEE Computer Society

July-September 2026

P.128-134

Abstract

Artificial intelligence (AI) is inescapable in scientific and engineering systems today. The adoption of AI features across these systems shifts our experiences in a variety of ways. Additionally, designing interfaces that include AI features is a rapidly shifting activity, in part because agentic AI is increasingly transforming the user experience of working with systems. This commentary explores work on human-centered AI and surfaces design guidelines for a range of forms of AI, including generative or agentic. These guidelines are relevant to IEEE Computing in Science & Engineering readers building a variety of systems. We conclude with a discussion of gaps for the community to consider in their system development endeavors, highlighting the need for AI to support collaboration in science.

Introduction

Artificial intelligence (AI) is inescapable in scientific and engineering systems today. AI adoption ranges widely, from using machine learning (ML) for detailed data processing and analysis to relying on commercially produced large language models to complete coding tasks. The current era of agentic AI is continually transforming work. Yet while transparency, reproducibility, security, and productivity dominate discussions of AI applications for science, talk of how these systems perform in the collaborative environments of research is largely lacking.

Design approaches used by the IEEE Computing in Science & Engineering (CiSE) community can benefit from work that describes human-centered design principles and methods for AI-enabled systems. Importantly, human–computer interaction (HCI) research, case studies, and resources from industry user experience (UX) provide frameworks and guidelines that should be used to inform the development of AI systems in science. For scientists and engineers, these recommendations help direct attention to key questions (e.g., What are our users’ relevant mental models? What do humans need to control and what factors influence that?) and offer patterns that can be applied when designing UXs. Leveraging these design approaches is not only helpful for creating more usable interfaces. Doing so is likely a key factor for creating systems that uphold principles of the scientific method as work expands to encompass even more automated systems.

Readers may question the likelihood that scientific software teams will adopt UX guidelines or methods, given spotty past adoption. We argue that the rapid adoption of agentic AI in scientific software development makes it easier to incorporate UX guidance, because these tools often invisibly incorporate such knowledge into their core models. However, we also assert that there is insufficient guidance on designing for collaboration between teams of people and AI systems. This is a significant and risky gap, with the potential to impede scientific productivity and innovation now and into the future. Continuing to build systems for isolated work is likely to slow team science and result in missed opportunities to expand involvement in cutting edge research.

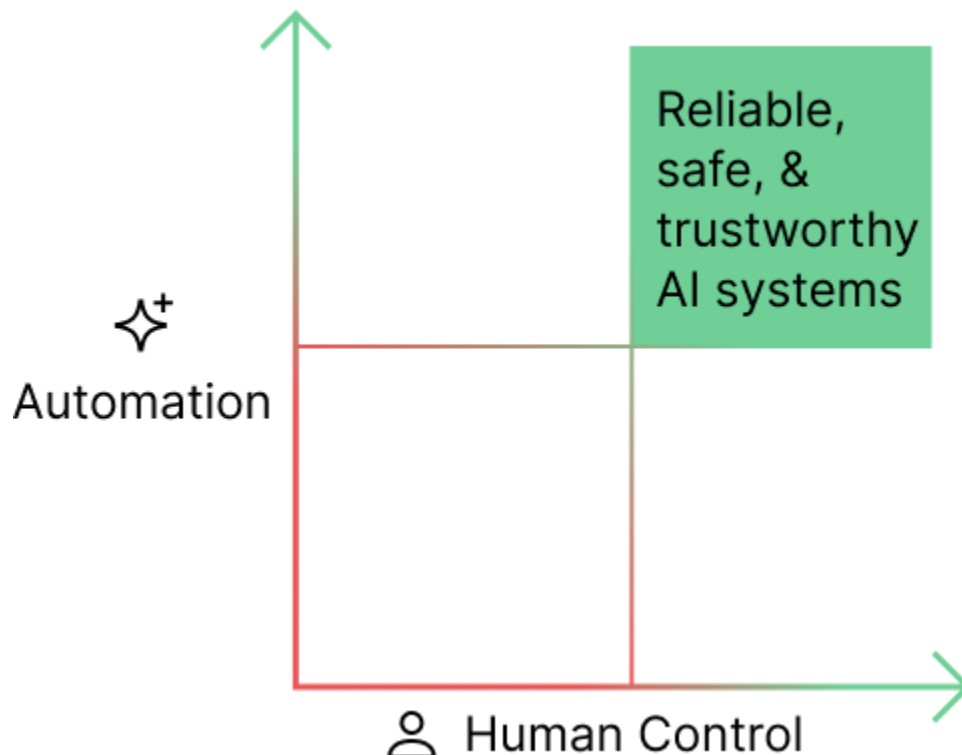
In this commentary, we introduce readers to principles and frameworks that can guide the thoughtful design of human-centered AI tooling. As we note below, however, these design guidelines alone cannot substitute for engaging in the UX work required to deliver contextually relevant systems for science, ones that are effective in collaborative environments. Research and engineering teams should carefully consider how to advance individual and team approaches to work in concert with advancing AI resources.

DESIGN PRINCIPLES AND AI FRAMEWORKS

The HCI community has spent decades exploring the human-centered design of AI methods, tools, and computational systems. Recent work organizes and summarizes design principles and frameworks for approaching the development of AI systems.

Shneiderman¹ for instance, describes a 2-D framework for what he calls “human-centered AI” (HCAI). In HCAI, designers determine how to balance human control against computer automation. Key to this HCAI framework is a claim that products need high amounts of human control combined with high automation to achieve reliable, safe, and trustworthy AI systems (see Figure 1). Shneiderman grounds this assertion with examples of self-driving cars and the evolution of airplane controls. We find that this view is reflected in the modern goal of matching an agentic AI coscientist with a “‘scientist-in-the-loop’ collaborative paradigm.”² The question is how to enable that high degree of human control and computer automation. This is still a matter of debate and discovery, highly reliant on context, and in scientific realms a subject of significant debate.

FIGURE 1. Author’s visualization of Shneiderman’s¹ framework for achieving reliable, safe, and trustworthy HCAI. He asserts achieving truly human-centered AI requires a high degree of both automation and human control.



Shneiderman offers eight principles for HCAI-designed interfaces:

- 1) *Overview first, zoom and filter, then details-on-demand*: provide high-level summaries and allow users to dig into areas of interest. For example, mapping applications start with a high level of zoom and allow a user to zoom in for more details on demand.
- 2) *Preview first, select and initiate, then manage execution*: outline an entire process as the first view then show progress as an activity is completed.
- 3) *Steer by way of interactive control panels*: provide human control via distinct panels.
- 4) *Capture history and audit trails from powerful sensors*: capture history for review, especially relevant in science where provenance is key to reproducible results.
- 5) *People thrive on human-to-human communication*: systems should enable users to share with other users, in addition to AI agents.
- 6) *Be cautious when outcomes are consequential*: enable oversight for mission-critical procedures, tools, and systems so that a human can sign off.
- 7) *Prevent adversarial attacks*: consider security concerns where malicious actors may attempt to leverage the system to cause issues.
- 8) *Incident reporting websites accelerate refinement*: be open to feedback and provide a clear avenue for receiving and then acting on it.

Shneiderman¹ emphasizes that AI should not cut out humans. He argues that design thinking around HCAI systems “will amplify, augment, empower, and enhance human performance.”

Weisz et al.³ assert their own set of design principles for developing generative AI applications. They argue that preexisting human–AI (and general) design guidelines do not “address the unique design challenges of generative AI” since such guidelines do not cover some key details.³ Weisz et al. believe current advice does not address the variability generative AI introduces, nor do they cover “new or amplified ethical issues stemming from the models’ generative nature.” To address these gaps, the authors conducted iterative user research resulting in six core design principles and strategies for building effective, usable generative AI-based applications. The first three reinterpret previously established concerns for designing AI applications. The last three focus specifically on newer issues arising with generative applications. The six principles defined are:

➤ *Design responsibly*: “Ensure the AI system solves real user issues and minimizes user harms.” *Design for mental models*: “Communicate how to work effectively with the AI system, considering the user’s background and goals.”

- › *Design for appropriate trust and reliance*: “Help the user determine when they should or should not rely on the AI system’s outputs by teaching them to be skeptical of quality issues, inaccuracies, biases, underrepresentation, and other issues.”
- › *Design for generative variability*: “Help the user manage the ability of generative models to produce multiple outputs that are distinct and varied.”
- › *Design for cocreation*: “Enable the user to influence the generative process and work collaboratively with the AI system.”
- › *Design for imperfection*: “Help the user understand and work with outputs that may not align with their expectations.”

With each principle, Weisz et al.³ provide strategies to help software teams design and implement a better experience. These vary widely, but examples include fundamentals like “use a human-centered approach” to understand user needs and “understand the users’ mental model.” Each of these are high-level strategies common in UX design of all types.

Common among Weisz et al.³ and Schneiderman’s¹ guidance is the idea that developers of AI systems must orient their design around the humans using those systems, empowering them and accounting for the new realities of generative AI. The guidelines and frameworks offered by these authors are a starting point for the CiSE community when designing AI systems or features. But core UX methods are needed to understand the complex contexts that users work in. By understanding users’ environments and everyday practices, we can align scientific software to such needs. This is also necessary to appropriately evaluate prototypes and tools for opportunities to incorporate new innovations. Guidelines give CiSE readers a framework for examining a problem and understanding what they need to build. But evaluations, such as those leveraging core heuristics or UX research methods, are also necessary (see “*Heuristic Evaluations of Interfaces*”).

While many UX methods involve engaging with users (e.g., through conversation), interfaces can be evaluated based on their alignment with design best practices. By comparing a system against best practices, we can identify usability concerns without needing to spend time recruiting participants. Nielsen’s 10 Usability Heuristics for User Interface Design⁴ articulate best practices for user interfaces. Adhering to their ideals reduces complexity for users and simplifies interaction with a system. By adopting this framework, evaluators can be less biased. Evaluators can systematically identify usability issues and come up with ways to address them.

Heuristic evaluations focus on how features are delivered, so that novice and expert users alike will think a tool is useful and easy to learn. Heuristic evaluations do not assess if a given feature is desirable by a user. They also do not comment on aesthetic choices unless they are likely to affect usability. Heuristic evaluations can be applied to graphical user interfaces, but

are appropriate for other interfaces as well (e.g., command line or application programming interface).

To conduct a heuristic evaluation, reviewers first familiarize themselves with Nielsen's 10 usability heuristics.⁴ Then, using a workbook like Nielsen Norman's or STRUDEL's,^{S1} which has been adapted for scientific software applications, reviewers can keep track of the issues they identify and the recommendations they have for addressing the problems. Prompt questions help reviewers recognize possible issues. For example, one heuristic (as adapted by STRUDEL) is Help Users Recognize, Diagnose, and Recover from Errors:

Error messages should be expressed in plain language, precisely indicate the problem, and constructively suggest a solution.

- *Does the design use traditional error message visuals, like bold, red text?*
- *Does the design offer a solution that solves the error immediately?*
- *Is there a way to monitor ongoing processes?*
- *Are log files available for reviewing issues?*
- *Is it clear when an issue is related to a third-party?*

In their workbook, a reviewer might log an issue related to this heuristic, possibly providing an example for clarity. An issue could be:

AI agent reports errors by referencing past queries but those queries are not named or IDed, making it ambiguous what is the object of discussion. For example: The agent says, "Let me check what went wrong with the query," and "Let me wait and check the status," but doesn't say which query is being checked.

Reviewers should provide clear recommendations to resolve the issue. These recommendations do not need to go so far as mocking up solutions or detailing a code fix but should provide additional clarity and identify a possible route forward. A recommendation for the above issue could be: *Provide IDs for queries so that they can be explicitly referenced.*

Issues that reviewers uncover are understood as possible occasions where usability can be hindered, though real-world users may not always encounter the anticipated frictions themselves.^{S2} Nevertheless, heuristic evaluations are a quick way to evaluate a UX. And, they do not require special data collection or analysis tools, only the system to be evaluated and a place to take notes.

REFERENCES

S1. D. Paine et al., “Heuristic evaluation template,” Zenodo, Jun. 2, 2026. [Online]. Available: <https://zenodo.org/records/20517042>

S2. R. W. Bailey, R. W. Allan, and P. Raiello, “Usability testing vs. Heuristic evaluation: A head-to-head comparison,” *Proc. Hum. Factors Soc. Annu. Meet.*, vol. 36, no. 4, pp. 409–413, 1992, doi: 10.1177/154193129203600431.

Industry Guidance

Alongside the principles and guidelines above, teams building commercial AI systems have researched and developed common guidelines and best practices. These insights frequently adapt academic work and reconfigure and evaluate it in the context of largescale deployed systems and services. Two readily available and applicable options for *CiSE* readers are from Google and Microsoft.

Google’s People and AI Guidebook

Google’s People + AI Research (PAIR) Guidebook⁵ offers advice on “designing for human-centered AI products.” Core insights are divided into multiple chapters: “User Needs and Defining Success,” “Data and Model Evolution,” “Mental Models and Expectations,” “Trust and Explanations,” “Feedback and Controls,” and “Errors and Graceful Failures.” This extensive guidebook offers a variety of advice and exercises that *CiSE* readers may find useful if interested in putting its guidance into practice.

Five key principles underlie the PAIR Guidebook’s advice. These principles are similar to those in both Shneiderman’s¹ HCAI framework and Nielsen’s heuristics.⁴ PAIR’s five key principles are:

- › *Design for the appropriate level of user autonomy:* Ensure that users can delegate or keep control as desired.
- › *Align AI with real-world behaviors:* Understand the context your tool will be used in and ensure user journeys, data collection, and model evaluation are aligned.
- › *Treat safety as an evolving endeavor:* Users will perform more complex tasks over time; ensure safety strategies can be adapted over time. Adapt AI with user feedback: Develop feedback mechanisms and use what you learn to refine the AI system.
- › *Create helpful AI that enhances work and play:* Ensure AI workflows allow users to do the kinds of work they enjoy.

Supplementing these principles are nearly two dozen patterns describing best practices for software teams designing AI applications. Some patterns set expectations for a development team’s approach, e.g., “determine if AI adds value.” While this may seem like common sense, in

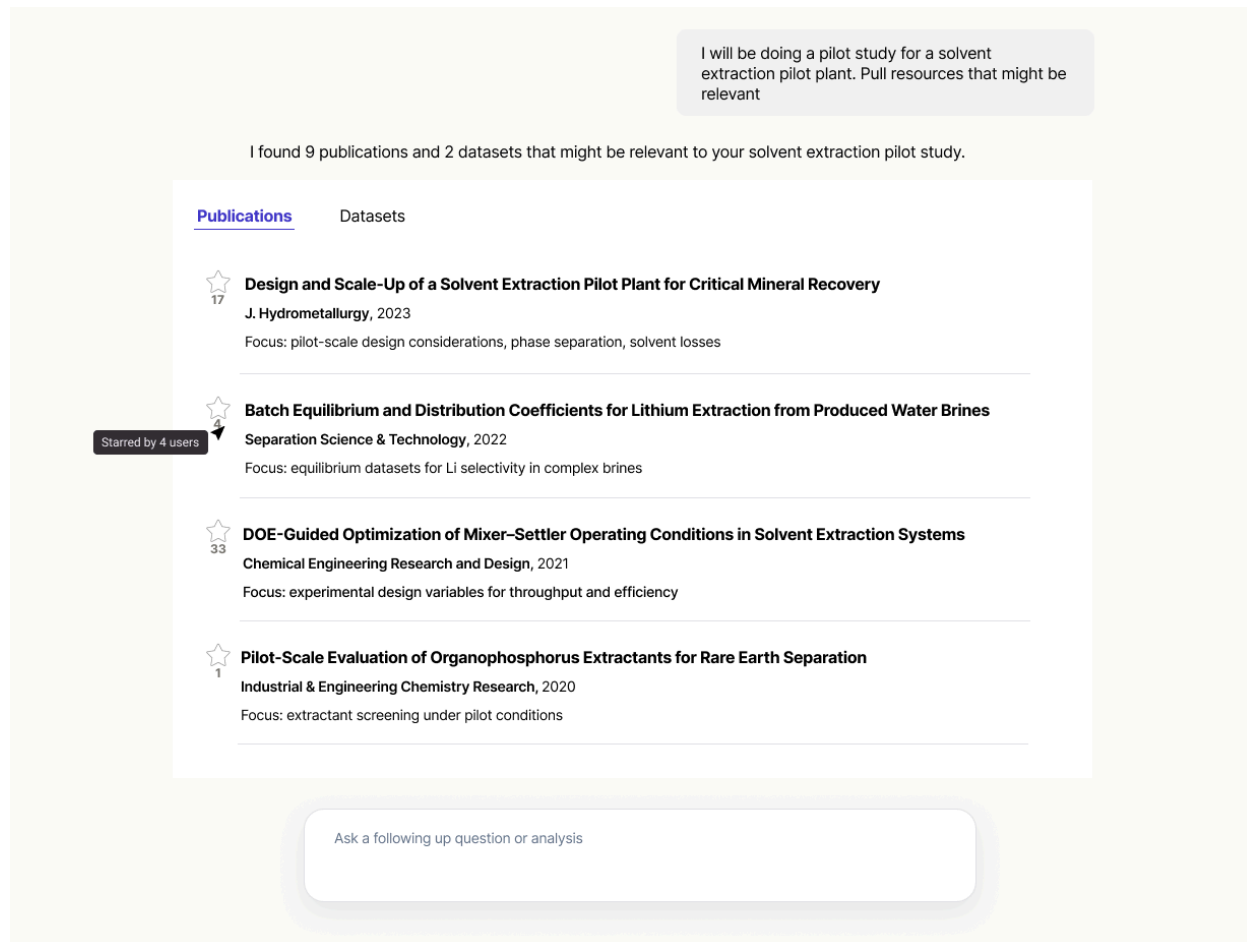
today's commercial market it is very easy to question whether many AI features do add value. In scientific software projects, a team may face pressure to integrate AI. Taking the opportunity to critically evaluate how or if adding an AI element really adds value to the scientific work in question is a worthwhile undertaking.

Other patterns in PAIR offer insight into how to design features, e.g., “add context from human sources.” This is relevant when building AI-enabled data search and exploration systems. For example, imagine a signal showing how popular any given resource in the result list is. Or, when searching for articles or datasets, like in *Figure 2*, being able to tell how many other searchers found a particular resource helpful via a counter showing the number of thumbs-ups, stars, or hearts users have awarded. Here, rather than relying solely on the AI system's rankings, users are able to judge utility based on other people's judgements. Additionally, user ratings are feedback to the AI system on the relevance of results to a query. This aligns with the fourth principle underlying the PAIR guidelines.

The adding context from human sources design pattern is also applicable beyond search results. When developing a workflow, or conducting analyses, researchers increasingly turn to AI agents that suggest a processing or analytical technique. Rather than expecting users to blindly trust the recommendation, AI agents can include links to research papers or grey literature that apply or explain the technique, allowing users to decide if it is good for their own use case.

Overall, PAIR design patterns often provide information and guidance on what to aim for and what to avoid. This is conceptually similar to the heuristics for evaluations (*as discussed in “Heuristic Evaluations of Interfaces”*), but they are meant to be applied at up front design time rather than when testing a prototype or functioning system.

FIGURE 2. The PAIR Guidebook offers the design pattern “add context from human sources.” When searching for resources, helpful context can include signals of popularity like counts of stars or hearts as shown in this image. Providing this information helps users make informed decisions, enabling the human control that Shneiderman¹ says is key to trustworthy AI systems. Image created by Sarah Poon and Hannah Cohoon.



Microsoft's HAX Toolkit

Microsoft similarly offers guidance on developing AI applications through its Human–AI eXperience (HAX) Toolkit.⁶ This is a resource for helping teams “conceptualize what the AI system will do and how it will behave.” Similar to the PAIR Guidebook, these guidelines are best used for up-front design rather than heuristic evaluation of existing solutions.

The HAX Toolkit provides¹⁸ guidelines organized according to when they play a strong role in the experience. This includes when a user initially encounters the AI system, during interaction with it, when something goes wrong, and over time as use continues. Eight of the HAX guidelines are associated with design patterns that show “flexible and actionable” solutions to problems that the guideline is meant to address. Design patterns each state the problem a user might have, a solution to it, and information on when, how, and why the solution is appropriate. Common pitfalls are also noted.

For instance, the guideline “Convey the consequences of user actions” can be implemented through the design pattern “Feedforward.” To feedforward in this context is to “convey the consequences of user actions before the user takes action.” The HAX Toolkit asserts that this pattern is best used when users have high-stakes consequences, e.g., there is no easy way to undo the action, and when users’ actions impact an AI’s decision making. A situation where this could be applicable is when an AI agent initiates a job on a supercomputer, using up the finite resources that the researcher was allotted. Rather than letting a job be launched with a single command, the agent could explain how long the job is expected to run and how many of their allotted node hours will be left when completed. After giving this information, the agent could ask for confirmation that the user wants to follow through.

The HAX Toolkit and PAIR Guidebook overlap somewhat in their content. For instance, HAX’s “encourage granular feedback” guideline is generally equivalent to PAIR’s “let users give feedback.” Much of the advice also aligns with Nielsen’s⁴ heuristics as well. Both HAX and PAIR have guidelines roughly equivalent to Nielsen’s “user control and freedom” principle, for example, which indicates that users should be able to correct mistakes easily. Despite similarities, HAX focuses more on user interaction than PAIR, which includes more advice on strategy for developing AI applications.

However, both design systems are missing critical information and resources that would make them more useful to scientific software. Neither system addresses the needs of dynamic human teams with respect to AI applications. Additionally, neither includes detailed examples of what applying the patterns or guidelines might look like. Without mockups or examples, *CiSE* readers may find it frustrating to try and use the guides. PAIR does present some products along with applicable patterns, but the specific implementations of those patterns within the products are left unsaid. HAX includes information on how to implement design patterns, but no demos or screenshots are included and descriptions are very high level. Both leave much to the reader’s imagination.

REFLECTIONS AND CONCLUSIONS

The *CiSE* community can reference the principles in Weisz et al.,³ Schneiderman,¹ and the PAIR Guidebook⁵ and HAX Toolkit⁶ when building AI-enabled tools. The dynamic nature of AI systems, and rapid advancement of the field itself, can make this seem like a daunting task. Yet the seemingly simple principles, strategies, or guidelines examined here can be used to improve the UX of a wide range of scientific and engineering tools.

One might question whether UX guidelines are likely to be widely adopted by designers of scientific AI tools. After all, UX is already underutilized in the resource-constrained environment of scientific software. Yet interestingly, the very fact that AI agents can leverage such design guidelines increases their importance. Humans may resist adopting best practices because of inertia, concern about their longevity, the cost or opportunity cost of adoption.⁷ AI agents lack these concerns and can readily integrate new instructions on how to craft outputs. Thus, while

UX approaches and design guidelines may be slowly adopted among scientific software developers themselves, the agents they dispatch to help create interfaces for still more AI-enabled science can adopt them immediately.

Yet available guidelines are not always well tailored to scientific contexts. Readers should consider: What do researchers and practitioners need to understand to design more usable AI/ML systems? Based on the work examined in this article, we assert that key questions to consider in science and engineering system design include:

- › What are users' mental models of the systems/ workflows that AI is impacting?
- › What cues do users need to determine the trustworthiness of information?
- › What do humans need to control and what factors influence that?
- › What design and development approaches align best with creating and maintaining AI systems?
- › What are effective design patterns for integrating AI into scientific user experiences?
- › How does collaboration among humans impact the design requirements of AI systems?

Addressing these questions is an activity UX practitioners and other CiSE community members should be equipped to explore. Some efforts are underway. For instance, drawing on user interviews, Cohoon et al.⁸ discuss how to align AI coscientist workflows with researchers' mental models. However, while many of the questions we propose align with the principles and patterns put forth by the guidelines we have reviewed, the final question regarding collaboration touches on an aspect largely ignored by existing advice. The work we have explored all positions AI-enabled work as something involving a single human. Despite the terms used to describe that work (e.g., "collaborative UX" and "cocreation") implying multitudes, existing advice on designing AI interfaces assumes that a solo user is at the helm. This assumption is pervasive. Gottweis et al.'s² concept of an AI coscientist, for example, is designed around a single user interacting with a multiagent system that provides scientific assistance. Feng et al.'s⁹ Cocoa is designed so that "a human user and an AI agent can jointly plan and execute plan steps," but makes no mention of how a scientific team might interact with the system. Additionally, Larsen-Ledet¹⁰ recently emphasized that generative AI tools are usually "restricted to conversational interfaces or, at best, confined to making things happen in a single app," which in practice "does not match how people work" in team environments.

While some scientific work may be solitary, a huge proportion involves working with ideas and artifacts in concert with other people, either synchronously or asynchronously. As scientists collect, process, synthesize, and interrogate data, we move it and insights across applications and storage sites. AI tooling must be designed to support a data lifecycle that takes place across complex tool assemblies, guided by the hands of multiple human collaborators. Visions

of “AI coscientists”² portray a soon to be real future where multiagent AI systems are creating hypotheses and executing research workflows to rapidly and autonomously conduct scientific experiments. Thus, a significant gap in the complex work of science and engineering is collaboration, between human–human teams as well as human–agent teams. Current guidelines and best practices for implementing AI interfaces discussed in this article address one human interacting with one or more AI agents. But, they are insufficient for addressing nuanced collaborative work among teams of humans and agents. Designing for such multifactor endeavors is an area worthy of attention from CiSE readers.

ACKNOWLEDGMENTS

The author's work at Lawrence Berkeley National Laboratory is supported by the U.S. Department of Energy Advanced Scientific Computing Research under Contract DE-AC02-05CH11231, the Alfred P. Sloan Foundation, and the U.S. National Science Foundation under Interagency Agreement #A2407-049-089-064206.0. The views expressed do not necessarily reflect the views of any funding agency or any other organization.

REFERENCES

1. B. Shneiderman, *Human-Centered AI*, 3rd ed. London, England: Oxford University Press, 2022.
2. J. Gottweis et al., “Accelerating scientific discovery with Co-Scientist,” *Nature*, vol. 655, no. 8122, pp. 1–3, 2026, doi: 10.1038/s41586-026-10644-y.
3. J. D. Weisz, J. He, M. Muller, G. Hofer, R. Miles, and W. Geyer, “Design Principles for Generative AI Applications,” in *Proc. CHI Conf. Human Factors Comput. Syst.*, Honolulu, HI, USA, vol. 10, 2024, pp. 1–22.
4. “How to conduct a Heuristic evaluation.” Nielsen Norman Group. Accessed: Sep. 14, 2023. [Online]. Available: <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>
5. “People + AI Guidebook.” People + AI Research. Accessed: Jun. 8, 2026. [Online]. Available: <https://pair.withgoogle.com/guidebook>
6. “Microsoft HAX Toolkit.” Microsoft. Accessed: Jun. 8, 2026. [Online]. Available: <https://www.microsoft.com/en-us/haxtoolkit/>
7. R. V. O'Connor and G. Coleman, “An investigation of barriers to the adoption of software process best practice models,” in *Proc. ACIS*, 2007, pp. 780–789.
8. J. Cohoon et al., “Toward calibrated trust: What working scientists need from an AI co-scientist,” *Zenodo*, 2026, doi: 10.5281/zenodo.20595967.
9. K. J. K. Feng et al., “Cocoa: Co-planning and coexecution with AI agents,” in *Proc. CHI Conf. Human Factors Comput. Syst.*, Barcelona, Spain, 2026, pp. 1–23.
10. I. Larsen-Ledet, “Generative AI at work: Mundane and productive articulation,” in *Proc.*

Extended Abstr. CHI Conf. Human Factors Comput. Syst., 2026, pp. 1–6, doi:
10.1145/3772363.3798602.

Author Bios

JOHANNA (HANNAH) COHOON is a user experience researcher in the Scientific Data Division at the Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA. Contact her at hcohoon@lbl.gov.

DREW PAINE is a lead user experience researcher in the Scientific Data Division at the Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA. Contact him at pained@lbl.gov.

