Designing Semi-Automated Copilots: Balancing User Control and Guidance for Effective Human-Agent Collaboration

Anjali Khurana
Computing Science
Simon Fraser University
Burnaby, BC, Canada
anjali_khurana@sfu.ca

Parmit K Chilana
Computing Science
Simon Fraser University
Burnaby, BC, Canada
pchilana@cs.sfu.ca

POSITION STATEMENT

The rise of Large Language Model (LLM)-based in-application assistants, such as copilots and LLM agents, presents both opportunities and challenges in balancing automation with human skill retention. While full automation can streamline repetitive tasks, over-reliance risks deskilling users and diminishing their sense of control, learnability and critical thinking. Our research explores how different automation paradigms- a fully automated copilot (AutoCopilot) and a semi-automatic copilot (GuidedCopilot)—affect user perceptions of learnability, user control and utility in feature-rich software. We found that semi-automatic approaches in GuidedCopilot, which provide step-by-step visual guidance while automating trivial steps, better support complex decision-making, debugging, and creative input. This approach shows potential in fostering users' mental models of the software, reducing cognitive overload, and enhancing skill retention. Our findings underscores the importance of balancing automation, guidance, and control for effective human-agent collaboration and highlight key factors to consider for designing GenAI systems that augment rather than replaces human decision-making. Our work aligns with key workshop questions, particularly in understanding what skills should be preserved vs offloaded to AI, and how to design AI assistants that enhance rather than replace human expertise. By participating, we hope to gain interdisciplinary insights into the theoretical foundations of human-AI collaboration and refine design principles that ensure AI serves as a supportive augmentation tool rather than a passive automation system.

CCS CONCEPTS

• **Human-centered computing → Graphical user interfaces**.

KEYWORDS

feature-rich software; large language models; software copilots; user control; semi-automation; human-agent collaboration

ACM Reference Format:
Anjali Khurana and Parmit K Chilana. 2025. Designing Semi-Automated Copilots: Balancing User Control and Guidance for Effective Human-Agent Collaboration. In . ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

CHI 2025 Workshop on Tools for Thought, Yokohama, Japan

https://doi.org/XXXXXXX.XXXXXXX

1 SUMMARY OF OUR WORK RELEVANT TO THE WORKSHOP

With the advancements in Generative AI (GenAI) and pre-trained Large Language Models (LLM) [1, 2, 6, 14], in-application GenAI-based assistants have become much more powerful and have enabled a new wave of assistants capable to automate a wide range of complex tasks. Such LLM-powered assistants are often branded as copilots or LLM agents [1, 13, 14], such as Microsoft 365 Copilot [14], Adobe Firefly [1], and Figma AI [9], setting the bar for user's expectation of what generative AI can do. These fully automated copilots intend to eliminate the efforts required for users to use feature-rich applications to complete their tasks. However, they introduce distinct challenges due to their open-ended nature, inherent complexities, and wide-ranging failure modes [4]. Furthermore, additional onus lies on users to review AI outputs and refine their inputs to align them with their intent, increasing overall effort [5, 15–17].

While there is a growing trend toward developing copilots that fully automate user tasks by bypassing intermediate steps, such end-to-end approaches provide users with minimal control over the task completion process. This not only deskills users, limiting their understanding of specific software features (e.g., how pivot tables function in spreadsheets), but also reduces their ability to perform tasks independently or intervene when automation fails. In contrast, HCI research on software help-seeking highlights that users prefer to *learn by doing* [7] when working with feature-rich applications. This presents a critical challenge in designing GenAI assistants: How can we create GenAI assistants that strike an optimal balance of automation, guidance, and user control, enhancing human cognition rather than replacing it?

In our recent CHI 2025 paper [12], we investigated what the concept of a copilot means to users and how GenAI copilots influence human skills and workflows within feature-rich software. Specifically, we examined two distinct automation paradigms [3, 8, 11]: whether users prefer copilots that fully automate tasks, as seen in fully automated systems, or those that serve as assistants, combining semi-automation for repetitive tasks with guided support and instructions for learning complex tasks. To study these design paradigms, we designed and implemented two in-application copilot interventions: AutoCopilot that fully automates software tasks (Figure 2), inspired by state-of-the-art copilot assistants (e.g., [9, 10, 14]); and GuidedCopilot, a novel semi-automatic copilot that automates only trivial or repetitive tasks while offering step-by-step visual guidance to help users locate UI elements (Figure 1). GuidedCopilot integrates visuals into its responses, allows users to initiate the automation process and enables corrections before

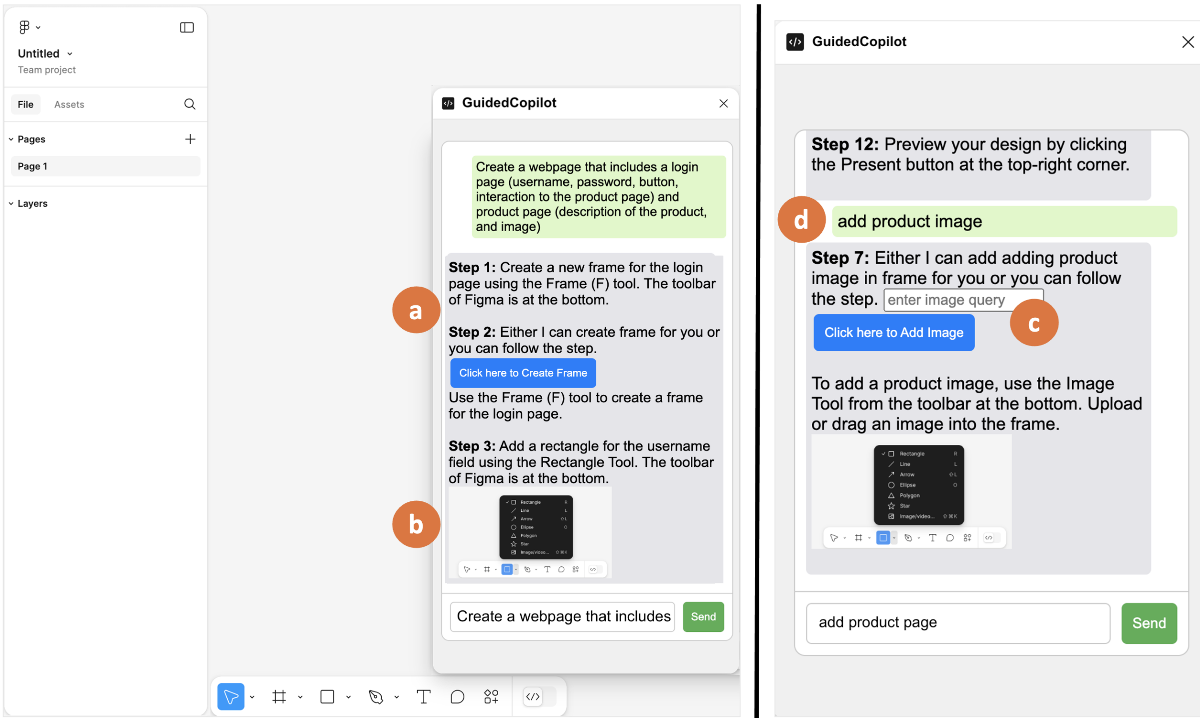

Figure 1: GuidedCopilot, a novel semi-automatic copilot: (a) Copilot assistance is structured to provide step-by-step guidance along with semi-automation for only repetitive or trivial steps in the task; (b) Visual references of the UI elements in-context to user's tasks and application are provided within the step-by-step guidance; (c) Users have control over editing the LLM extracted entities from their query before the semi-automation is performed; (d) Up-to-date mixed-medium follow-up responses are provided. To see the contrast with the fully automated copilot assistance, please see AutoCopilot in Figure 2.

proceeding. Our study investigates how these how these design paradigms (semi-automation vs. full automation) impact task completion, user perception of control, utility and learnability, and augment their capabilities—considering factors such as user expertise (novice vs. expert), familiarity with LLMs, and task nature (fixed vs. creative/exploratory).

We conducted a within-subject controlled experiment (N=20) and follow-up interviews to compare the strengths and weaknesses of AutoCopilot vs. GuidedCopilot across data analysis and visual design tasks. Our findings highlight that automating repetitive and trivial tasks, such as done by GuidedCopilot, can be effectively offloaded to AI-driven copilots, allowing users to focus on higher-value cognitive and creative activities. However, fully automated systems can negatively impact users' perceptions of software learnability and user control. The lack of explainability within fully automated systems hinders users from building a mental model of software features or learn how to manipulate automation. Our study suggests that high levels of automation are best suited for simple, repetitive tasks and workflows, while tasks requiring complex decision-making, debugging, validation, or creative problem-solving (e.g., intricate data analysis or exploratory design) benefit more from semi-automation, where GuidedCopilot provides visual step-by-step guidance to augment human skills while allowing users to initiate automations and enable corrections to retain control. To ensure effective human-AI collaboration, we synthesized the insights from our studies into key factors and levels of automation along with guidance to consider in feature-rich software environments.

2 DESIGN AND IMPLEMENTATION OF OUR INTERVENTIONS: AUTOCOPILOT AND GUIDEDCOPILOT

We designed AutoCopilot (Figure 2) to provide full automation for the entire software task and offer concise follow-up responses based on context from the software documentation and web data, similar to in-application state-of-the-art copilots. For implementing AutoCopilot, we used a hybrid-RAG using OpenAI's text-embedding-3 model [63] and Elasticsearch [1, 73]. Our GuidedCopilot (Figure 1) reflected semi-automated systems and provided semi-automation for repetitive or trivial steps along with visual step-by-step guidance. GuidedCopilot allowed users to initiate semi-automation and edit LLM extracted entities before proceeding. We used GraphRAG, GPT-4o, BAAI/bge-base-en embedding model and LLM Agent to implement GuidedCopilot. Both copilots occasionally generated errors and exhibited uncertainty, when its HybridRAG or GraphRAG fails to identify relevant semi-automation functions. for the detected intent. Both the copilot UIs module was built as an in-application assistant and migrated to Chrome as an extension for Google Sheets and as a plugin for Figma.

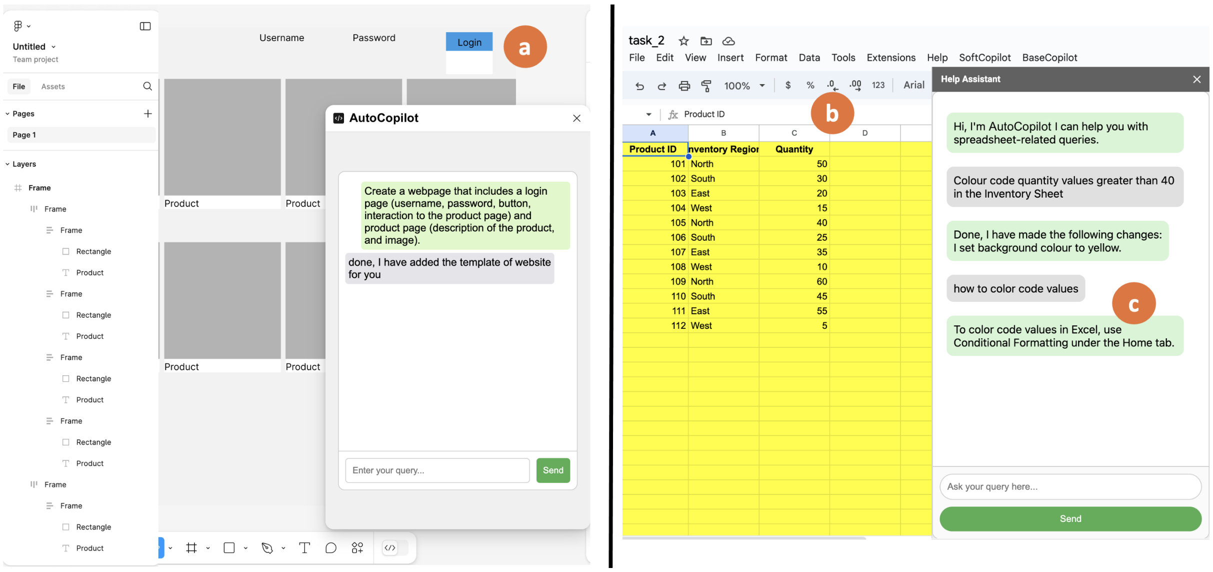

Figure 2: AutoCopilot: (a) Fully automates the user's task (e.g., creating a webpage that includes a login and product page); (b) Similar to state-of-the-art copilots, demonstrates incorrect automation (such as color coding the entire sheet instead of values greater than 40 in column C); (c) Provides follow-up textual response based on context from software documentation

3 KEY TAKEAWAYS

3.1 Balancing user control and guidance

Our experience in designing and studying these paradigms of copilots provide compelling evidence that users value maintaining control when interacting with software copilots, particularly for complex or unfamiliar tasks. By challenging the trend toward fully automated copilots, our findings underscore the importance of balancing automation, guidance, and control for effective human-agent collaboration. This aligns with theories of human-computer interaction (HCI) and mixed-initiative systems, where AI assists rather than replaces human decision-making. The study also highlights the importance of adaptive, step-by-step visual guidance, which supports users in developing a mental model of the software, reducing cognitive overload, and enhancing retention. GenAI systems should therefore prioritize explainability, control, guidance, along with automation to support diverse user needs, ensuring that human skills are augmented rather than diminished.

3.2 Key factors to consider for levels of automation

The design of GenAI systems presents both opportunities and challenges in protecting and enhancing human thinking. One key opportunity lies in developing AI-driven assistants that dynamically adjust to user expertise and task complexity, ensuring that automation supports rather than replaces critical thinking. However, a major challenge is preventing users from becoming passive monitors, a risk associated with full automation. Our study synthesizes three key factors for determining the appropriate level of automation and user control in software copilots: (i) Familiarity with the application: When designing copilots for unfamiliar applications, a guided, semi-automatic approach with visual references can help users onboard, while more adaptive support for experts balances time savings with user control, catering to different expertise levels. (ii) Task Type:Higher levels of automation are best suited for straightforward, repetitive, or simpler visual tasks. For more nuanced tasks involving complex decision-making, debugging, user dependencies, or creative input, semi-automatic copilots with step-by-step visuals or previews are more beneficial, allowing for greater user control. (iii) User Intent in learning vs. performing tasks: For users with a clear intent to learn while using feature-rich software, future copilots should adopt a semi-automatic approach with step-by-step guidance. For those focused on time efficiency, such as expert software users and CS male users, a higher level of automation would be more suitable.

Our findings provides empirical evidence that copilots should function truly as "co-pilots" [74], supporting users without diminishing their role or skills, and draw from established help-seeking strategies [14, 27, 28, 40] to maintain a balance between automation and user learning.

REFERENCES

[1] Adobe. 2023. Adobe unveils Firefly, a family of new creative generative ai. https://news.adobe.com/news/news-details/2023/Adobe-Unveils-Firefly-a-Family-of-new-Creative-Generative-AI/default.aspx

[2] Open AI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt

[3] Lisanne Bainbridge. 1983. Ironies of automation. In *Analysis, design and evaluation of man–machine systems*. Elsevier, 129–135.

[4] Gagan Bansal, Jennifer Wortman Vaughan, Saleema Amershi, Eric Horvitz, Adam Fourney, Hussein Mozannar, Victor Dibia, and Daniel S. Weld. 2024. *Challenges in Human-Agent Communication*. Technical Report MSR-TR-2024-53. Microsoft. https://www.microsoft.com/en-us/research/publication/human-agent-interaction-challenges/

[5] Shraddha Barke, Michael B. James, and Nadia Polikarpova. 2023. Grounded Copilot: How Programmers Interact with Code-Generating Models. *Proc. ACM Program. Lang.* 7, OOPSLA1, Article 78 (apr 2023), 27 pages. https://doi.org/10.1145/3586030

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda

Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[7] John M. Carroll and Mary Beth Rosson. 1987. *Paradox of the Active User.* MIT Press, Cambridge, MA, USA, 80–111.

[8] Mica R Endsley. 2017. From here to autonomy: lessons learned from human–automation research. *Human factors* 59, 1 (2017), 5–27.

[9] Figma. 2024. Figma AI: Your Creativity, unblocked with Figma AI — figma.com. https://www.figma.com/ai/. [Accessed 11-09-2024].

[10] Adobe Firefly. 2024. https://firefly.adobe.com/

[11] Eric J. Horvitz. 2007. Reflections on Challenges and Promises of Mixed-Initiative Interaction. *AI Magazine* 28, 2 (Jun. 2007), 3. https://doi.org/10.1609/aimag.v28i2.2036

[12] Anjali Khurana, Xiaotian Su, April Yi Wang, and Parmit K Chilana. 2025. Do It For Me vs. Do It With Me: Investigating User Perceptions of Different Paradigms of Automation in Copilots for Feature-Rich Software. In *CHI Conference on Human Factors in Computing Systems (CHI '25)* (Yokohama, Japan). ACM, New York, NY, USA, 18. https://doi.org/10.1145/3706598.3713431

[13] Abigail Sellen and Eric Horvitz. 2024. The rise of the AI Co-Pilot: Lessons for design from aviation and beyond. *Commun. ACM* 67, 7 (2024), 18–23.

[14] Jared Spataro. 2023. Introducing Microsoft 365 copilot – your copilot for work. https://blogs.microsoft.com/blog/2023/03/16/introducing-microsoft-365-copilot-your-copilot-for-work/

[15] Ashley Stewart. 2024. A CIO canceled a Microsoft AI deal. The reason should worry the entire tech industry. — businessinsider.com. https://www.businessinsider.com/pharma-cio-cancelled-microsoft-copilot-ai-tool-2024-7. [Accessed 11-09-2024].

[16] Priyan Vaithilingam, Tianyi Zhang, and Elena L. Glassman. 2022. Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 332, 7 pages. https://doi.org/10.1145/3491101.3519665

[17] Litao Yan, Alyssa Hwang, Zhiyuan Wu, and Andrew Head. 2024. Ivie: Lightweight Anchored Explanations of Just-Generated Code. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 140, 15 pages. https://doi.org/10.1145/3613904.3642239