Maintenance of Personal Generative User Interfaces

ZHICHENG HUANG, University of California San Diego, United States
HAIJUN XIA, University of California San Diego, United States

Generative user interfaces promise on-demand customization and support for long-tail tasks. Through examples, we show that generated interfaces may create maintenance burdens at both individual and collaborative levels. We frame these challenges as an infrastructure problem and outline directions for supporting the long-term coordination and evolution of generative interfaces.

CCS Concepts: • **Human-centered computing → Interaction paradigms**.

Additional Key Words and Phrases: generative user interfaces, personal information management

1 Introduction

The rapid advance of AI models is driving a resurgence of generative user interfaces (GenUIs), which promise to support long-tail tasks through on-demand customization. Prior work largely focuses on innovation, improving generation techniques and enabling richer personalization. We argue that as GenUIs become more ubiquitous, their flexibility introduces a less examined challenge: maintenance.

We focus on *personal generative user interfaces*, GenUIs designed for or by end users rather than professional designers [3, 14]. Within this space, we distinguish between ephemeral interfaces for ad-hoc lookup tasks [11] and persistent interfaces used for longitudinal or recurring work. While the former are short-lived, the latter integrate into everyday life and accumulate maintenance demands as both tasks and representations evolve.

We examine how such personal GenUIs create maintenance burdens at both individual and collaborative levels, and discuss the infrastructural implications of supporting their long-term evolution and coordination.

2 Maintenance Burdens of Personal GenUIs

GenUIs make it easy to generate task-specific interfaces on demand, supporting long-tail needs but shifting effort toward longitudinal maintenance. This burden appears at two levels: *personal*, where individuals manage evolving interface ecologies, and *collaborative*, where shared understanding must be sustained across divergent systems.

Personal maintenance: Sally, a student, relies on generated interfaces throughout daily life. She captures quick thoughts in a minimal note interface, logs workouts in a structured tracker, manages projects in a media-rich editor, annotates lecture slides in a class interface, and later generates a synthesis interface to study for exams. Each interface is locally useful, but over time she must move ideas from notes into structured trackers, reformat project artifacts for other contexts, and keep her study view aligned with evolving materials. The difficulty is not losing information, but maintaining coherence across multiple interfaces that encode different structures and assumptions. She must maintain not just data, but the relationships between evolving interface logic.

Collaborative maintenance: Mike tracks symptoms and daily routines with a generated personal health interface, using narratives and shorthand meaningful to him. In a clinical visit, a clinician accesses the same data through her own GenUI composed of standardized diagnostic fields aligned with her clinical workflow. Although the underlying data is shared, its representations diverge: what Mike refers to as "a recurring pattern" may not correspond cleanly to the

Authors' Contact Information: Zhicheng Huang, University of California San Diego, La Jolla, California, United States, zhichenghuang@ucsd.edu; Haijun Xia, University of California San Diego, La Jolla, California, United States, haijunxia@ucsd.edu.

clinician's view, and references such as "that episode" or "last week's notes" require extra effort to align. Maintenance here involves repairing shared grounding across heterogeneous GenUIs rather than simply transferring data [13].

3 Infrastructure Implications for Maintaining Personal GenUIs

Rather than asking people to reconcile fragmented artifacts, we discuss the infrastructural implications of maintaining the GenUIs that produced them and the evolving schemas they embody. Treating interfaces as part of the representational medium—not just tools—helps meanings persist and transform across personal and collaborative contexts.

Schema evolution: Supporting long-tail personal GenUIs requires persisting not only artifacts but also the schemas and generative logic that structure them. Interfaces become part of the representational history of artifacts. This view builds on Personal Information Management work that treats digital personal space as a database-like substrate spanning documents, tools, screens, and web content [5–8, 15]. We extend this trajectory by framing personal GenUIs and their generation processes as an evolving schema layer. Interface evolution thus becomes schema evolution [4], requiring infrastructure to track dependencies and rule-based relationships (e.g., Datalog [2] queries) to maintain tractability and synchronization.

Synchronization: Supporting personal GenUIs over time requires synchronization across devices and collaborators. The state of the art for shared artifacts relies on Conflict-Free Replicated Data Types (CRDTs), which ensure deterministic merging and syntactic convergence under opportunistic, eventually consistent synchronization [12, 16, 17].

However, interfaces can be generated in many ways, and not all of them are equally compatible with this synchronization model. In unguided generation, where interfaces are produced directly from informal natural-language prompts, interface logic is often embedded in arbitrary code or loosely structured artifacts. This makes meaningful changes dependent on semantic understanding of how data is structured and interpreted, and independently generated interfaces may not even share the same schema or vocabulary. In such cases, syntactic merging alone is insufficient.

A generation process with intermediary specifications [1, 9, 10] offers more tractability. When interface structure and behavior are expressed through explicit schemas or intermediate specs, key semantics are reflected in the representation itself, making synchronization more manageable.

4 Future Work

We conclude by outlining research directions that emerge from viewing personal GenUIs as an ongoing problem of maintenance, where interfaces, schemas, and their relationships evolve over time.

- How should end-user–facing environments be designed to support ongoing maintenance of evolving interfaces and their dependencies, including making schema relationships visible and allowing users to manage interface evolution over time?
- How can intermediate specifications be designed so that schemas can be derived directly from them, balancing expressiveness with tractability?
- As different platforms adopt intermediate GenUI specifications with varying semantics, how can conflicts across heterogeneous schema logics be resolved beyond syntactic CRDT-style merging?

References

[1] Yining Cao, Peiling Jiang, and Haijun Xia. 2025. Generative and Malleable User Interfaces with Generative and Evolving Task-Driven Data Model. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–20.

[2] Stefano Ceri, Georg Gottlob, Letizia Tanca, et al. 1989. What you always wanted to know about Datalog(and never dared to ask). *IEEE transactions on knowledge and data engineering* 1, 1 (1989), 146–166.

[3] Xiang'Anthony Chen, Tiffany Knearem, and Yang Li. 2025. The GenUI Study: Exploring the Design of Generative UI Tools to Support UX Practitioners and Beyond. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference*. 1179–1196.

[4] Carlo A Curino, Hyun J Moon, and Carlo Zaniolo. 2008. Graceful database schema evolution: the prism workbench. *Proceedings of the VLDB Endowment* 1, 1 (2008), 761–772.

[5] Paul Dourish, W Keith Edwards, Anthony LaMarca, and Michael Salisbury. 1999. Presto: an experimental architecture for fluid interactive document spaces. *ACM Transactions on Computer-Human Interaction (TOCHI)* 6, 2 (1999), 133–161.

[6] Susan Dumais, Edward Cutrell, Jonathan J Cadiz, Gavin Jancke, Raman Sarin, and Daniel C Robbins. 2003. Stuff I've seen: a system for personal information retrieval and re-use. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. 72–79.

[7] Jim Gemmell, Gordon Bell, and Roger Lueder. 2006. MyLifeBits: a personal database for everything. *Commun. ACM* 49, 1 (2006), 88–95.

[8] Zhicheng Huang and Jeff Huang. 2025. IRCHIVER: An Information-Centric Personal Web Archive for Revisiting Past Online Sensemaking Tasks. In *Proceedings of the 2025 ACM SIGIR Conference on Human Information Interaction and Retrieval*. 231–246.

[9] Saketh Ram Kasibatla, Kiran Medleri Hiremath, Raven Rothkopf, Sorin Lerner, Haijun Xia, and Brian Hempel. 2025. The Command Line GUIde: Graphical Interfaces from Man Pages via AI. In *2025 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 173–178.

[10] Alan Leung, Ruijia Cheng, Jason Wu, Jeffrey Nichols, and Titus Barik. 2025. SQUIRE: Interactive UI Authoring via Slot QUery Intermediate REpresentations. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–17.

[11] Yaniv Leviathan, Dani Valevski Matan Kalman, Danny Lumen, Eyal Segalis Eyal Molad, Shlomi Pasternak, Vishnu Natchu, Valerie Nygaard, and Srinivasan Cheenu Venkatachary James Manyika Yossi Matias. [n.d.]. *Generative UI: LLMs are Effective UI Generators*. Technical Report. Technical report, Google Research, 2025. Available at generativeui. github. io.

[12] Geoffrey Litt, Sarah Lim, Martin Kleppmann, and Peter Van Hardenberg. 2022. Peritext: A crdt for collaborative rich text editing. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–36.

[13] Pavel Okopnyi, Oda Elise Nordberg, and Frode Guribye. 2024. Against Generative UI. In *Proceedings of the Halfway to the Future Symposium*. 1–4.

[14] Seokhyeon Park, Soohyun Lee, Eugene Choi, Hyunwoo Kim, Minkyu Kweon, Yumin Song, and Jinwook Seo. 2026. Bridging Gulfs in UI Generation through Semantic Guidance. *arXiv preprint arXiv:2601.19171* (2026).

[15] Dennis Quan, David Huynh, and David R Karger. 2003. Haystack: A platform for authoring end user semantic web applications. In *International semantic web conference*. Springer, 738–753.

[16] Hyun-Gul Roh, Myeongjae Jeon, Jin-Soo Kim, and Joonwon Lee. 2011. Replicated abstract data types: Building blocks for collaborative applications. *J. Parallel and Distrib. Comput.* 71, 3 (2011), 354–368.

[17] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. 2011. Conflict-free replicated data types. In *Symposium on Self-Stabilizing Systems*. Springer, 386–400.