

From Logs to Agents: Reconstructing High-Level Creative Workflows from Low-Level Raw System Traces

TAE HEE JO, Design Informatics Lab, Hanyang University, Republic of Korea and Human-Centered AI Design Institute, Hanyang University, Republic of Korea

KYUNG HOON HYUN*, Design Informatics Lab, Hanyang University, Republic of Korea and Human-Centered AI Design Institute, Hanyang University, Republic of Korea

Current AI-based Creativity Support Tools (CSTs) generate massive amounts of low-level log data (e.g., clicks, parameter tweaks, metadata updates) that are hard to interpret as "creative intent". We argue that to enable future agentic systems to understand and assist users, we must first translate these noisy system traces into meaningful high-level user behavioral traces. We propose a method that parses raw csv/JSON logs into structured behavioral workflow graphs that map the provenance and flow of creative assets. By abstracting low-level system events into high-level behavioral tokens (e.g., MODIFY_Prompt, GENERATE_Image), this method enables downstream analyses like sequence mining and probabilistic modeling. We discuss how this structured workflow history is a prerequisite for 'Process-Aware Agents'—systems capable of suggesting next design moves or explaining rationales based on a deeper understanding of the user's workflow patterns and history.

Additional Key Words and Phrases: creative activity traces, creativity support tools, design process analysis, workflow, agent enhancement

ACM Reference Format:

Tae Hee Jo and Kyung Hoon Hyun. 2026. From Logs to Agents: Reconstructing High-Level Creative Workflows from Low-Level Raw System Traces. 1, 1 (February 2026), 5 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 Introduction

The landscape of Creativity Support Tools (CSTs) has shifted from linear editing software to complex, node-based generative environments such as ComfyUI [1], Flora AI [2] and GENPRESSO [6] While these tools offer high creativity, they present a significant challenge for both researchers and system designers: the "black box" of the creative process. While we possess rich raw data in the form of csv/JSON system logs, and theories of creativity such as divergent versus convergent thinking [3, 5], we lack a computational representation of the user's creative journey that is both machine-readable and semantically meaningful.

The gap is technical and semantic. Raw logs from node-based CSTs can be extremely noisy. In our analysis of generative design sessions, we observed that a single "creative design move" such as tweaking a prompt to change a lighting effect, can generate dozens of system events, including connection re-routing, system cleanup routines, and

*Corresponding author.

Authors' Contact Information: Tae Hee Jo, persontae94@gmail.com, Design Informatics Lab, Hanyang University, Seoul, Republic of Korea and Human-Centered AI Design Institute, Hanyang University, Seoul, Republic of Korea; Kyung Hoon Hyun, hoonhello@gmail.com, Design Informatics Lab, Hanyang University, Seoul, Republic of Korea and Human-Centered AI Design Institute, Hanyang University, Seoul, Republic of Korea.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

redundant metadata updates. Analyzing creativity at this resolution is inefficient, as it is both computationally difficult and cognitively loading [4].

We propose a workflow reconstruction pipeline that automates the translation of low-level system trace logs into high-level user behavioral workflow graphs. We argue that this restructuring data process is necessary to enhance the next generation of CST agents. Current agents operate on state (the current canvas); to be truly collaborative, they must understand the full workflow process history (how the user arrived there).

2 The Proposed Method

We present a three-stage pipeline designed to transform raw csv/JSON logs into actionable user behavioral insights. This method has been applied and tested on generative design logs from a node-based CST (GENPRESSO).

2.1 Semantic Filtering and Classification (De-noising)

The first challenge in analyzing raw trace data is distinguishing explicit high-level user intent from low-level system noise. Raw data logs often contain echoes of user actions which are redundant internal server calls, state synchronization events, and temporary object lifecycle markers that do not reflect a new creative decision.

We implement a heuristic classification layer that filters events based on `Action_Type` and the system’s `Raw_Source_Label` to isolate meaningful creative moves.

- **Filtering System Noise:** Our pipeline automatically discards events that represent system-side maintenance rather than user intent. This includes cleanup events where the system purges temporary node or asset IDs, purely structural updates related to node connection logic (e.g., backend graph re-routing), and intermediate state updates that occur during asynchronous generation processes. By removing these artifacts, we prevent the analysis from interpreting system latency or housekeeping as user hesitation or activity.
- **Heuristic Classification:** We classify the remaining high-intent events into four creative design moves based on a rule-based mapping of the raw logs. **INSERT:** Represents the introduction of new content into the canvas space. This includes importing external images, creating new nodes; **MODIFY:** Captures the refinement of existing assets. This includes altering context within a prompt, image, or video nodes, or adjusting the generation output parameters (e.g., AI model, aspect ratio); **GENERATION:** Distinct from simple insertions, this classification is reserved for the execution of the generative AI model itself, marking the moment a user commits their parameters to produce a new artifact (image or video); **REMOVE:** Represents the explicit deletion of assets, distinguishing between a user rejecting a generated outcome and the system clearing a temporary cache.

In our pilot dataset, this filtering layer reduced the event volume by approximately 40% (from 927 raw system logs to 563), successfully isolating sequences that represent tangible changes to the creative artifact while discarding backend redundancy.

2.2 Design Sequence Reconstruction (Visualizing the workflow history)

Creativity in generative tools is rarely linear [7]; it is branching. A user may generate an image, branch off to try a variation, abandon it, and return to the parent node. Timestamps alone are insufficient to track this evolution within the user’s creative design process, as they only record *when* an action happened, not *where* it fits in the design ideation hierarchy.

We construct a Directed Acyclic Graph (DAG) (1) by resolving parent-child relationships extracted from the raw metadata source fields (e.g., `connected_from` fields). The X-axis represents the generation depth (distance from the log start) and Y-axis represents the chronological order of log events specific to its generation depth. Each node contains unique number labels indicating the sequence number of each user’s creative design move; black number indicates the global sequence order, blue number indicates the sequence of the MODIFY design move, and red number indicates the sequence of the REMOVE design move. Nodes represent creative assets (Prompt Nodes, Image Processes, Video Outputs) and edges represent the flow of data. Square nodes indicate an AI generated output, while circle nodes indicate a user’s manually created creative asset.

We utilize a depth-based layout algorithm (implemented in Python/NetworkX) to visualize the creative design session. As shown in Figure 1, this reveals the overall shape of the design process workflow of the user’s CST design session, distinguishing between wide exploration (many branches) and deep refinement (long branches).

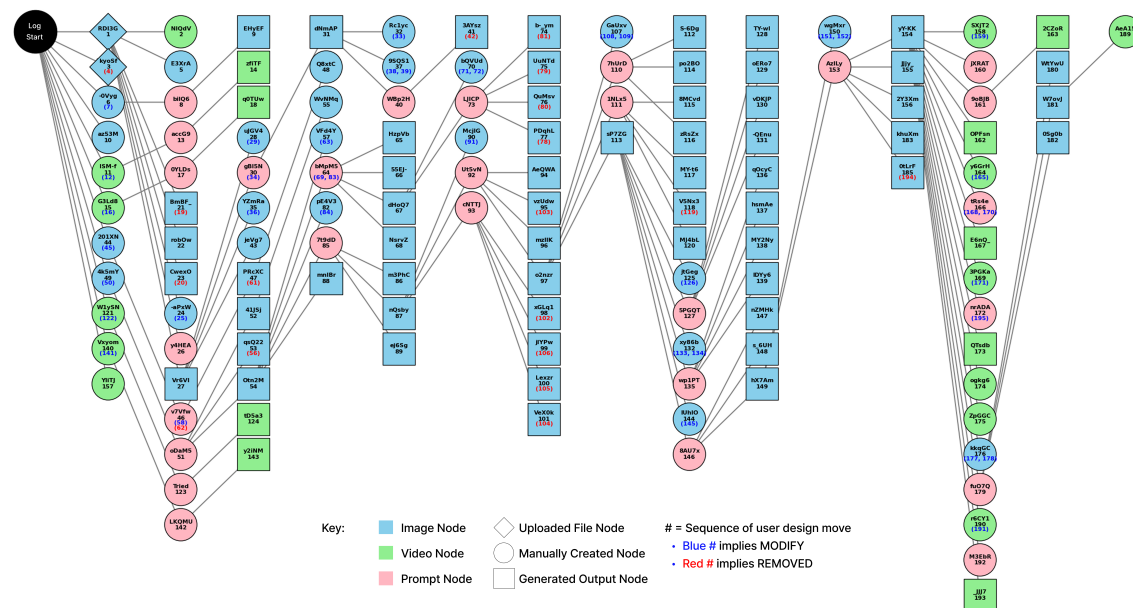


Fig. 1. This abstract user behavioral workflow graph visualizes the branching evolution of a creative design process session. Blue nodes indicate image nodes, green nodes indicate video nodes, and pink nodes indicate prompt nodes. The structure reveals a user exploring multiple parallel variations before converging on a single lineage.

2.3 Tokenization

To make these graphs analyzable, we convert the nodes into standardized tokens. We concatenate the user action and node asset type to create a standardized language of interaction such as, `INSERT_prompt`, `MODIFY_image`, `GENERATION_video`. This tokenization abstracts away the specific tool interface, allowing for potential cross-platform comparison.

3 Implications: Enhancing Process-Aware Agents

While the primary contribution of this work is the reconstruction pipeline of low-level system traces to high-level user behavioral workflow itself, its value lies in the downstream applications for Agentic AI for CSTs.

3.1 From Descriptive to Predictive Analysis

By converting logs into behavioral tokens, we can move from describing what happened to modeling the probability of what happens next. In our pilot sequence analysis of 563 log events, we identified distinct probabilistic signatures. For instance, The most common Bigram (2-step sequence) was GENERATION_image - GENERATION_image (19.1%, Frequency=37). This indicates a high frequency of 'blind' re-rolling, suggesting the user is relying on the stochastic nature of the model rather than refining their input.

To further quantify these behaviors, we applied Markov Chain analysis to model state transition probabilities [8]. We found that after a user inserts an image node (INSERT_image), there is a 69.6% probability that their immediate next action is to modify its content (MODIFY_image). Conversely, after generating an image, there is a 66.1% probability of immediate re-generation. These transition probabilities provide a mathematical basis for an agent to distinguish between "setup phases" (high probability of modification) and "exploration phases" (high probability of re-generation).

3.2 Scaffolding Agentic Decisions

We argue that the user behavioral workflow history graph data should be fed into the context window of future CST agents (e.g., LLM-driven assistants), as an agent cannot effectively collaborate if it has no memory of the workflow history. For instance, if a user has performed the sequence INSERT_image - MODIFY_metadata_update (resizing image) five times in a row, current agents would wait for the user's next command. However, a process-aware agent, aware of the high transition probability (69.6%) between insertion and modification, could then infer the intent and offer *"I see you are importing and resizing references. Shall I automatically apply this scale factor to future imports?"* Therefore, by analyzing the workflow history graph data, an agent can provide better rationale. Instead of saying *"Try this prompt,"* it could suggest smarter decision making moves *"Users who followed a similar generation path found success by {pattern context}."*

4 Conclusion

We proposed a workflow reconstruction pipeline to turn low-level raw system data into high-level, actionable behavioral insights. By filtering noise, classifying raw data, reconstructing design sequence lineage, and tokenizing actions, we provide a structured foundation for analyzing user behavioral design process workflows based off raw log data in node-based CSTs. Beyond analysis, this structured data serves as a critical enabler for the next generation of AI agents. We argue that by equipping agents with a "memory" of the user's behavioral design process history, we can transition from passive tools that merely execute commands to process-aware collaborators. Such agents could leverage historical workflow patterns to predict user intent, suggest strategic design moves, and provide rationale grounded in the provenance of the creative process.

Acknowledgments

This work was supported by the Industrial Technology Innovation Program(RS-2025-02317326, Development of AI-Driven Design Generation Technology Based on Designer Intent) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

References

- [1] Comfy Org. 2024. ComfyUI. Website. <https://www.comfy.org/> Accessed: 2026-02-09.
- [2] FLORA. 2025. Flora.ai. Website. <https://www.florafaua.ai/> Accessed: 2026-02-09.

- [3] Gabriela Goldschmidt. 2014. *Linkography: Unfolding the Design Process*. MIT Press, Cambridge, MA.
- [4] Shilin He, Xu Zhang, Pinjia He, Yong Xu, Liqun Li, Yu Kang, Minghua Ma, Yining Wei, Yingnong Dang, Saravanakumar Rajmohan, et al. 2022. An empirical study of log analysis at Microsoft. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1465–1476.
- [5] Seung Won Lee, Tae Hee Jo, Semin Jin, Jiin Choi, Kyungwon Yun, Sergio Bromberg, Seonghoon Ban, and Kyung Hoon Hyun. 2024. The Impact of Sketch-guided vs. Prompt-guided 3D Generative AIs on the Design Exploration Process. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM. doi:10.1145/3613904.3642218
- [6] Recon Labs Inc. 2025. GENPRESSO. Website. <https://genpresso.ai/> Accessed: 2026-02-09.
- [7] Amy Smith, Barrett R. Anderson, Jasmine Tan Otto, Isaac Karth, Yuqian Sun, John Joon Young Chung, Melissa Roemmele, and Max Kreminski. 2025. Fuzzy Linkography: Automatic Graphical Summarization of Creative Activity Traces. In *Proceedings of the 2025 Conference on Creativity and Cognition*. ACM. doi:10.1145/3698061.3726915
- [8] Harold Thimbleby, Paul Cairns, and Matt Jones. 2001. Usability analysis with Markov models. *ACM Transactions on Computer-Human Interaction (TOCHI)* 8, 2 (2001), 99–132.