# How Structure Mediates Creative Activity Traces

Vincent Cavez
vcavez@stanford.edu
Stanford University
Stanford, USA

## Abstract

Creative activity traces captured through software creativity support tools are shaped not only by user behavior but also by the structural behavior of the tools themselves. We argue that without accounting for how tool structure constrains, filters, and transforms user actions into recorded events, trace analysis risks conflating structural mediation with creative intent. Drawing on a framework that characterizes structure along two dimensions, rigidity (how much the user can shape the rules) and enforcement (how much the user is bound by the rules during interaction), we show that rigidity controls the completeness of a trace (what categories of creative activity it can record) and enforcement controls its fidelity (how faithfully it captures moments where user intent and structural rules conflict). We propose that trace analysis should account for the structural behavior of the tools that produce those traces, and outline research directions toward this goal.

## Keywords

structure, flexibility, malleable interfaces, interaction design, theoretical framework

**ACM Reference Format:**
Vincent Cavez. 2026. How Structure Mediates Creative Activity Traces. In . ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 Introduction

Research on creative activity traces has made significant progress in recent years. Linkographs and their extensions capture the emergence and structure of design moves [1, 2]. Interaction logs from creativity support tools provide fine-grained records of how users explore, produce, and revise [3]. These traces are increasingly used to evaluate creativity support tools, for instance by measuring how AI-based assistance affects the diversity of creative output [4], and recent work has begun mapping the design space of creative activity traces themselves [5].

Yet creative activity traces are never raw. They are always captured through software, and that software imposes structure on what users can do and, consequently, on what traces can record. A composer sketching a rhythmic phrase in a notation editor, a DAW, and a freeform canvas will produce three qualitatively different traces, not because they behaved differently in each, but because each tool's structure filtered the interaction differently: the notation editor rejected invalid input, the DAW let some actions bypass the beat grid while constraining others, and the canvas imposed no structural constraints at all.

We argue that this structural mediation is an underexamined confound in trace analysis, and that accounting for it requires a vocabulary for describing tool structure and its behavior. The Structural Interaction framework [9] provides such a vocabulary. It models the user interface as a set of rules that stand between the user and their content, and places these rules at the center of interaction design. Crucially, it characterizes rule behavior along two orthogonal dimensions, rigidity (how much the user can shape the rules) and enforcement (how much the user is bound by the rules during interaction). We show that rigidity shapes the completeness of a trace, determining whether the user's structural adaptation is visible or invisible in the record, while enforcement shapes its fidelity, determining how the trace renders moments where user intent and structural rules come into conflict.

## 2 Background: Structure and Its Behavior

The Structural Interaction framework [9] models a user interface as a directed graph of elements and rules. Elements are the entities users perceive: a note on a musical staff, a layer in a compositing tool, a clip on a timeline. Rules are the organizational logic that determines how elements relate, behave, and appear. Structure is the set of all rules governing the organization of elements. The user triggers rules, rules transform elements, and the user perceives the result. Within any structure, closely related rules cluster into sub-structures that users experience as a unit: the beat grid in a DAW, the notation grammar in a score editor, or the layer hierarchy in a compositing tool.

Sub-structures can be characterized along two orthogonal dimensions. **Rigidity** captures how much the user can shape the rules, from ***fixed*** (immutable), to ***negotiable*** (adjustable within bounds), to ***malleable*** (entirely redefinable), to ***authorable*** (the user creates rules from scratch). **Enforcement** captures how much the user is bound by the rules during interaction, from ***persistent*** (the rule does not yield), to ***elastic*** (yields under pressure but restores itself), to ***escapable*** (tolerates bypass that persists), to ***liftable*** (the rule can be removed entirely). These two dimensions are orthogonal: any rigidity value can combine with any enforcement value, generating a 4×4 design space of 16 combinations [9].

## 3 Structure Shapes What Traces Can Capture

We now develop each point in turn, showing how each value of rigidity and enforcement produces a distinct trace signature.


*Conference'17, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

### 3.1 Rigidity and Trace Completeness

Rigidity determines what categories of user activity are visible in the trace. As rigidity decreases, an increasing share of the user's structural work becomes part of the record.

***fixed*: only within-structure activity is visible.** When rules are immutable, traces capture content creation, element manipulation, and navigation, but nothing about how the user adapts to structural constraints. In a notation editor where musical rules are ***fixed***, the trace records note entries, deletions, and edits. It cannot record alternative arrangements the composer considered but never tried, or ideas they dismissed because the tool's grammar would not accommodate them. The structural constraint is invisible in the trace precisely because it is absolute.

***negotiable*: parameter adjustments become visible.** When rules are adjustable within bounds, each adjustment is a trace event. In a DAW, quantization strength is typically ***negotiable***: the user can set it to 100%, 75%, 50%, or off, but cannot redefine what quantization means. A composer who progressively loosens quantization as a piece develops is making a structural decision that a ***fixed*** tool would render invisible.

***malleable*: rule redefinitions become visible.** When the user can redefine rules entirely, the trace captures a richer form of structural work. A digital artist who redefines a brush preset in Procreate (adjusting size dynamics, opacity curves, texture) is not merely producing strokes but crafting the instrument through which strokes are produced.

***authorable*: rule creation becomes visible.** When the user creates rules from scratch, the trace captures the most complete picture of creative activity. A composer who builds a custom Max/MSP patch to generate harmonic accompaniment, or a designer who writes a Figma plugin to automate layout operations, is authoring the structural environment itself. These acts of rule creation are often the most revealing indicators of expertise and creative strategy, and they are entirely absent from traces produced by tools at lower rigidity levels.

The consequence for trace analysis is direct: studies that compare creative processes across tools with different rigidity levels are comparing traces of fundamentally different completeness. The difference in trace richness reflects a difference in structural configuration, not necessarily a difference in creative behavior.

### 3.2 Enforcement and Trace Fidelity

Enforcement determines how traces record the moments when user intent and structural rules come into conflict. Different enforcement values produce qualitatively different trace signatures from the same underlying creative impulse.

***persistent*: conflict produces absence.** The user either attempts an action that the rule silently blocks, or anticipates the rejection and never acts at all. In both cases, nothing is recorded. In commercial music notation software, musical rules (pitch spelling, rhythmic alignment, voice leading) form a sub-structure whose dominant state is (***fixed***, ***persistent***): they admit no deviation and do not yield during interaction. Cavez et al. [7] documented how this configuration hinders exploration in score writing: composers wishing to sketch ideas or try alternatives must comply with full notation enforcement or abandon the digital tool entirely. From a trace analysis perspective, every intention that the user abandons before acting on it, or that the system silently blocks, is absent from the trace. A composer who considers five different rhythmic arrangements but only enters the last one leaves the same trace as a composer who writes the final version directly.

***elastic*: conflict produces a transient deviation.** The user pushes against a constraint, the element moves, then snaps back automatically. In the same notation software, positioning rules (element placement, bar spacing) form a sub-structure whose dominant state is (***fixed***, ***elastic***): magnetic snapping resists manual adjustment but yields under sustained pressure before restoring default alignment [9]. In an interaction log, this appears as a brief spatial excursion that resolves itself. The creative impulse that motivated the push is visible only as a transient fluctuation, easily lost in noise if the analysis does not specifically look for it.

***escapable*: conflict produces a lasting trace.** The user bypasses the rule and the result persists. In a DAW like Logic Pro, the beat grid is ***fixed*** but its enforcement is ***escapable***: real-time recording places notes wherever the performance lands, and manual dragging moves MIDI notes or regions past the grid's magnetic resistance to arbitrary positions. The grid can even be disabled entirely (***liftable***), but this is rarely desirable since magnetic snapping remains useful for most editing. In practice, users work within ***escapable***: they escape the grid selectively rather than removing it. The resulting timing deviations, a slightly late snare, a deliberately off-grid region boundary, persist in the data and reveal exactly where user intent diverged from structural logic.

***liftable*: conflict produces a regime change.** The user removes a constraint entirely, and from that point forward, the trace is produced under a different structural configuration. In Cavez et al.'s work on pen-based score writing [8], the enforcement of the positioning sub-structure was shifted from ***elastic*** to ***liftable***: elements move freely during pen manipulation, and alignment constraints are not automatically restored but reapplied by the composer when the passage is ready. If a trace analysis does not account for this transition, it may interpret post-lifting behavior as if the alignment constraints were still active, leading to incorrect conclusions about the user's relationship to structure.

### 3.3 The Interaction of Rigidity and Enforcement

The two dimensions interact. For instance, platform migrations can shift enforcement without changing rigidity, altering what traces capture while leaving the tool's apparent identity unchanged. This has been documented in spreadsheets, where the grid's enforcement shifts from ***escapable*** to ***persistent*** when migrating from desktop to tablet, making it difficult to access values inside cells [6]. In the creative domain, the same pattern applies: a DAW whose beat grid is ***escapable*** on desktop (real-time recording, manual note dragging, and region arrangement all bypass it) may become ***persistent*** on a simplified mobile version (input is auto-quantized, regions snap to fixed positions). The trace of a performance on the mobile version contains none of the timing deviations or off-grid placements that would have been recorded on desktop, not because the performer played differently, but because the structural configuration no longer permits escape.

Conversely, Cavez et al. [8] addressed the limitations of commercial notation software by moving each sub-structure independently: the rigidity of musical rules shifted from ***fixed*** to ***negotiable*** within designated canvas zones, and the enforcement of positioning rules shifted from ***elastic*** to ***liftable***. Together, these moves transform the trace from a sparse record of compliant actions into a richer account of exploratory behavior.

## 4 Implications for Trace Analysis

These observations lead to a central research question: *how should trace analysis account for the structural behavior of the tools that produce those traces?*

We see three concrete research directions.

**Structural metadata for traces.** When researchers collect creative activity traces, they could document the structural configuration of the tool: which sub-structures are present, what their rigidity and enforcement values are, and whether these values change during the recorded session. A trace from a music notation editor could be annotated with the fact that musical rules are (***fixed***, ***persistent***) and positioning rules are (***fixed***, ***elastic***), immediately signaling that notation intentions that the user abandons or that the system blocks are absent from the trace while positioning deviations appear as transient fluctuations.

**Cross-structural comparison.** The design space provides a basis for controlled comparisons. A study where participants perform the same creative task in tools with systematically varied enforcement or rigidity configurations would isolate the effect of structural behavior on trace characteristics. Existing trace analysis methods such as fuzzy linkography [2] could serve as dependent measures in such studies, revealing how structural configuration affects the patterns that automated analysis detects. The platform migration pattern, where the same tool shifts enforcement across devices, offers natural experiments for such comparisons.

**Trace-aware tool design.** If certain structural configurations systematically impoverish traces (***persistent*** hides abandoned or blocked intentions, ***fixed*** hides structural adaptation), then tool designers who care about process visibility might deliberately choose configurations that produce richer traces. A creativity support tool designed for research could offer ***escapable*** rather than ***persistent*** enforcement not because escapability is always better for the user, but because it produces traces that better reflect creative intent. The quality of the trace is itself a design parameter, and the structural behavior of the tool is the mechanism through which designers can control it.

## 5 Conclusion

Creative activity traces are always produced through tool structure, and tool structure is not neutral. Different structural configurations filter, transform, and suppress different aspects of creative behavior. Rigidity controls how much of the user's structural adaptation is visible in the trace; enforcement controls how faithfully the trace records moments where user intent and structural rules collide. We believe that making this mediation explicit, through a vocabulary that characterizes how structure behaves, is a necessary step toward trace analysis methods that can distinguish what users did from what the tool allowed them to do.



## References

[1] Gabriela Goldschmidt. 2014. *Linkography: Unfolding the Design Process*. MIT Press.
[2] 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 17th Conference on Creativity & Cognition (C&C '25)*.
[3] Max Kreminski, Melanie Dickinson, Noah Wardrip-Fruin, and Michael Mateas. 2022. Loose Ends: A Mixed-Initiative Creative Interface for Playful Storytelling. In *Proceedings of the 8th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE '22)*.
[4] Barrett R. Anderson, Jash Hemant Shah, and Max Kreminski. 2024. Homogenization Effects of Large Language Models on Human Creative Ideation. In *Proceedings of the 16th Conference on Creativity & Cognition (C&C '24)*.
[5] Noor Hammad, David Chuan-En Lin, Amy Smith, Max Kreminski, Erik Harpstead, and Jessica Hammer. 2026. Tracing Creativity: A Design Space For Creative Activity Traces in HCI. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26)*.
[6] Vincent Cavez, Caroline Appert, and Emmanuel Pietriga. 2024. Spreadsheets on Interactive Surfaces: Breaking through the Grid with the Pen. *ACM Trans. Comput.-Hum. Interact.* 31, 2, Article 16, 33 pages.
[7] Vincent Cavez, Catherine Letondal, Emmanuel Pietriga, and Caroline Appert. 2024. Challenges of Music Score Writing and the Potentials of Interactive Surfaces. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Article 728, 16 pages.
[8] Vincent Cavez, Catherine Letondal, Caroline Appert, and Emmanuel Pietriga. 2025. EuterPen: Unleashing Creative Expression in Music Score Writing. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Article 760, 16 pages.
[9] Vincent Cavez, Kashif Imteyaz, and Anne-Flore Cabouat. 2026. Structural Interaction: Shifting the Focus of User Interface Design. Under review.