The Lucid Developer: Using LLM Hallucination as a Tool for Thought in Software Specification

LUCID: Leveraging Unverified Claims Into Deliverables

Ty Wells | Independent Researcher | ty@snapperland.com

CHI 2026 — TOOLS FOR THOUGHT WORKSHOP

GitHub: github.com/gtsbahamas/hallucination-reversing-system • DOI: 10.5281/zenodo.18522644

PROBLEM / MOTIVATION

AI coding assistants (Copilot, Cursor, Claude Code) treat hallucination as a **failure mode**. Developers manually inspect AI output, discarding or fixing hallucinated content. This places the full cognitive burden of verification on the developer.

Three independent formal proofs establish that hallucination **cannot be eliminated** from LLMs — it is mathematically intrinsic (Xu 2024, Banerjee 2024, Karpowicz 2025).

> *If hallucination is inevitable, fighting it is the wrong strategy.*
> **What if we harness it instead?**

KEY INSIGHT

Hallucination is computationally identical to the brain's **pattern completion** mechanism. In predictive processing (the dominant framework in cognitive neuroscience), all perception is "controlled hallucination" (Seth 2021, Clark 2023).

The brain generates predictions, then constrains them with sensory data. LLMs do the same — they generate predictions, constrained (or unconstrained) by context.

> **LUCID** *treats hallucination the way the brain treats its own predictions: generate freely, then constrain iteratively.*

THE TOOL: LUCID

Leveraging Unverified Claims Into Deliverables — a 6-phase iterative cycle:

1. **Describe** — Provide deliberately incomplete app description
2. **Hallucinate** — LLM writes Terms of Service for the app
3. **Extract** — Parse each claim into a testable requirement
4. **Build** — Implement using claims as acceptance criteria
5. **Converge** — Verify claims against actual codebase
6. **Regenerate** — Feed verified reality back to model

Why Terms of Service? Legal language forces precision across every dimension: functionality, security, privacy, performance, operations, compliance. A single hallucination produces 80–150 testable claims in 30 seconds.

CONVERGENCE RESULTS

Applied to a production Next.js app (~30K lines, 200+ files):

ITERATION	COMPLIANCE SCORE
1	**~35% (est.)**
3	**57.3%**
4	**69.8%**
5	**83.2%**
6	**90.8%**

91	5	$17	90.8%
CLAIMS EXTRACTED	CATEGORIES	TOTAL COST	FINAL SCORE

Claim categories: Functionality (34), Security (18), Data Privacy (16), Operational (12), Legal Compliance (11). 5 remaining FAILs = genuine missing features, not false positives. Monotonic convergence (score never decreases).

HOW THIS IS A "TOOL FOR THOUGHT"

LUCID reframes the developer's cognitive role:

CURRENT PARADIGM	LUCID PARADIGM
"Is this AI-generated code correct?" *(line-by-line verification — exhausting, error-prone)*	"What does the AI think my app should be?" *(specification-level exploration — higher abstraction)*

The developer becomes a **lucid dreamer**: participating in AI hallucination with metacognitive awareness, harvesting useful specifications while maintaining the ability to distinguish generated from real.

This is Engelbart's "augmenting human intellect" applied to the hallucination problem: the tool augments specification thinking by **exploiting, not suppressing**, the AI's generative tendency.

DESIGN PRINCIPLES FOR HALLUCINATION-AWARE TOOLS

> **1. Separate generation from verification**
> Don't constrain the generative process. Let the model confabulate freely, then apply external checking.

> **2. Use document formats as cognitive forcing functions**
> Different formats force different kinds of specificity. ToS forces breadth and precision simultaneously.

> **3. Design for iterative convergence, not one-shot correctness**
> Following predictive processing: perception is a rapid predict-compare-update loop. AI tools should support the same.

WHAT'S NOVEL

- ✓ No existing tool deliberately exploits hallucination for specification generation
- ✓ Only methodology combining AI-generated specs + deliberate hallucination + iterative convergence
- ✓ Neuroscience grounding provides design principles, not just metaphors
- ✓ Open source: working CLI tool on GitHub (DOI: 10.5281/zenodo.18522644)

LIMITATIONS & FUTURE WORK

Limitations:

- Claim quality depends on model training data
- Verification uses LLM (not formal verification)
- Tested on one production application so far
- Developer experience needs formal HCI evaluation

Future Work:

- User studies comparing developer cognition under LUCID vs. conventional AI-assisted specification
- Measuring specification coverage, cognitive load, and metacognitive awareness
- Multi-document hallucination (API docs, privacy policies, compliance certs)
- CI/CD integration for continuous specification drift detection
- Application beyond software: regulatory compliance, safety certification

DISCUSSION QUESTIONS

> *1. How might "hallucination as cognitive resource" change how we design AI development tools?*

> *2. What other document formats could serve as productive hallucination vehicles?*

> *3. Can the lucid dreaming metaphor inform interaction design beyond software?*

> *4. How do we measure whether a tool genuinely augments developer thinking vs. just automating a task?*

THEORETICAL GROUNDING

NEUROSCIENCE	LUCID DESIGN
Predictive processing: perception = controlled hallucination	Hallucinate freely, then constrain with verification
Confabulation: generation & checking are separable (Hirstein 2005)	LLM generates; codebase verification checks
Lucid dreaming: metacognition within generation (Baird 2019)	Developer navigates hallucinated space with awareness
System 1 / System 2 (Kahneman 2011)	LLM = System 1; LUCID verification = System 2
Protein hallucination: neural net dreams as blueprints (Nobel 2024)	Software hallucination: LLM dreams as specifications

REFERENCES

Anishchenko et al. 2021. De novo protein design by deep network hallucination. *Nature*.

Baird et al. 2019. The cognitive neuroscience of lucid dreaming. *Neurosci. & Biobehav. Rev.*

Banerjee et al. 2024. LLMs Will Always Hallucinate. arXiv:2409.05746.

Clark 2023. *The Experience Machine*. Pantheon.

Friston 2010. The free-energy principle. *Nature Rev. Neurosci.*

Hirstein 2005. *Brain Fiction*. MIT Press.

Huang et al. 2024. LLMs Cannot Self-Correct Reasoning Yet. *ICLR*.

Kahneman 2011. *Thinking, Fast and Slow*.

Karpowicz 2025. On the Fundamental Impossibility of Hallucination Control in LLMs. arXiv:2506.06382.

Seth 2021. *Being You*. Dutton.

Xu et al. 2024. Hallucination is Inevitable. arXiv:2401.11817.