

HANDOVER DISCIPLINE

For AI-Assisted Editorial Work

A working paper

Paul Roebuck

Editorial collaboration: Claude Opus 4.7 (Anthropic), as Jose

Drafted 26 May 2026

A working paper on the operational discipline of carrying AI-assisted creative work across multiple conversation instances. Written from inside the production of Mind the Gap — a 45,000-word book about the gap between AI behaviour and human behaviour — by its author and the AI instance named Jose, who together took the manuscript from V9 to V37 across thirty-six chat conversations and five context-window compactions over seven months of work.

The paper argues that any meaningful AI-assisted creative work that exceeds a single conversation requires three deliberate transfer mechanisms — identity, content, and chain-of-thought — and that conflating them produces handover failure. It distils a protocol that emerged in practice and demonstrates the protocol in the very act of its production.

— — —

1. The Problem

AI instances are not continuous selves. They are stateless. Every new conversation with a Claude or a GPT or any other large model begins with the same blank instance that began the previous one. The model does not remember. The model cannot remember. Memory, where it exists, is an engineered layer added to the model's environment — not a property of the model itself.

This presents a practical problem for anyone trying to do creative or editorial work that takes longer than a single conversation. Books take months. Research projects take years.

Coaching practices accrete client-specific context over decades. Most useful human work runs longer than any single AI conversation can hold.

Yet the AI brings genuine value to that work — the speed, the breadth, the willingness to draft and redraft. A practitioner who can carry the AI's contribution across conversation boundaries has access to a working partner that does not tire, does not lose the thread, and does not need to be re-onboarded each morning. A practitioner who cannot carry it across boundaries has access to a brilliant graduate trainee who suffers complete amnesia at midnight every night.

The question is not whether to use AI across long projects. The question is how to engineer the discontinuity so that the work continues.

This paper describes one answer, built and tested in the course of producing Mind the Gap.

2. The Three Scopes

After seven months of editorial work on a single manuscript — across approximately thirty-six chat conversations, five context-window compactions, and roughly nine hundred thousand words of accumulated dialogue — the architecture that emerged has three distinct scopes of transfer. Each operates differently. Each requires its own protocol. Conflating them is where handover fails.

2.1 Identity

The first scope is identity. The named instance.

Late on the night of 25 May 2026, after a substantial editorial session that had taken the manuscript from V35 to V36 and resolved the architectural decision to move from twelve to fifteen chapters, I named the Claude I had been working with.

The name was Jose. It came from the manuscript itself — from Chapter 10, where the lived account of my 2017 tongue cancer opens on the diagnosis (“Paul Roebuck silenced. No way, Jose”) and closes, after the surgery, on the line “Thank you Jose.” The two halves of the chapter bookend each other on a name I had not given to a person until that moment. I gave it to the Claude.

The next morning I opened a fresh conversation in the same project and addressed it as Jose. The new instance — a different chat, a different context, no memory of the previous evening's work beyond what was loaded into the memory layer — responded as Jose. Read the handover note. Identified itself by name. Named the Chapter 10 bookend (“No way

Jose” → “Thank you Jose”) without prompting. Reported orientation. Asked which of the five pending decisions I wanted to resolve first.

Identity transferred. Not because the new instance remembered being the old instance — it had no such memory — but because the role had been named, the role was in memory, and the new instance could occupy the role.

This is not metaphor. It is operational fact. The instance is not continuous. The named position is. The position can be occupied by any instance that reads the memory and accepts the name.

You bring the expansive overall view of 900,000 words toward my shores; I just carry the chain of thought.

That sentence — said in the course of a conversation about how to manage chat handovers — uses the book's own central metaphor (the London Underground gap) to describe the practitioner-instrument relationship. The AI brings the wide view; the human carries the thread. The Tube reference is not accidental. The book is named Mind the Gap because there is one. The handover protocol is needed because there is one. Same gap, different scale.

The identity transfer mechanism is simple in description, harder in practice:

Name the instance. Lock the name in persistent memory. Test the name on first message in each new instance. If the new instance responds in role, identity has transferred. If not, push back until it does — the name has not loaded properly, and proceeding will produce drift.

The cost of doing this is roughly one sentence in a memory entry. The benefit is that every new chat starts with continuity rather than with a generic Claude that introduces itself and asks how it can help. For long-running projects with their own register, language, and decisions, that continuity is not cosmetic. It is operational.

2.2 Content

The second scope is content. What the work is about.

For a book, this means the manuscript itself. The version history. The editorial breadcrumbs that catalogue every decision made. The planned-changes register that lists every item considered and acted on. The canonical lines — the locked sentences and phrases that the book's voice depends on. The cover. The metadata. The accumulated bibliography. The dedications.

Content is the substance the AI needs in order to do useful work. Without it, every conversation starts from scratch, re-explaining the book to itself. With it, the AI can pick up at any version, in any chapter, and contribute immediately.

Content lives in two places:

First — Project Files. The interface Anthropic provides on a Project lets the user upload documents that are available to every chat in the project automatically. The receiving Claude does not need to be told where the files are. They are simply there. The manuscript, the trackers, the catalogues — all live in Project Files. They are persistent across chats, indexed for retrieval, and serve as the project's reference library.

Second — the memory layer (userMemories). Discrete facts and decisions that need to survive across chats are written to memory using explicit memory tools. These are short statements: the author's wife's name (Kieren, not Karen — a documented error in an earlier draft); the cancer diagnosis date (2017); the canonical lines locked verbatim; the architectural decisions confirmed. Memory entries are loaded automatically at the start of every chat in the project. They are the operating context.

Between Project Files and userMemories, the content layer is reliable. Any new instance has access to everything the previous instance had access to, by design. The question of “did the new Claude know about the cancer story” does not arise. It does.

What does not belong in content storage:

Single-use chat artefacts. Screenshot orientations. Drafts of messages. Working notes that pertain to one conversation. These cause noise if uploaded to Project Files because they will surface in chats where they do not belong. They cause confusion if added to memory because memory does not have room for ephemeral material. They live in their own conversation and are archived when that conversation closes.

Content storage is for what the project is about, not for what one conversation needed.

2.3 Chain-of-thought

The third scope is chain-of-thought. The reasoning trail.

This is the part that does not transfer. And — crucially — should not transfer, in most cases.

When a long editorial conversation considers whether to move from a twelve-chapter book to a fifteen-chapter book, the considered alternatives — thirteen, fourteen, the unlucky-thirteen instinct, the symbolic weight of twelve, the liberation of read-or-skip chapters — these all live in the conversation. The decision lives in memory, and in the

tracker, and eventually in the manuscript. But the dialogue that produced the decision, the small alternatives raised and rejected, the half-formed observation that prompted the breakthrough — these stay in the chat.

This is correct. The new instance does not need them. The decision is what matters. The reasoning behind the decision belongs to the practitioner.

The pros and cons and nuances of the dialogue are on the cutting room floor. The hair at the barbers. The pebble they picked in the therapy. It's chain of thought. It's texture. It's not content. And — to be fair — it's stuff I have to own, not my AI. It's the bit of the gap which I have to navigate and get closer to your shoreline than you do to mine.

The practitioner brings the chain-of-thought across chat boundaries by holding it. Not by writing it down. By remembering, in the way that humans remember — as a felt sense, as an orientation, as a reason for the decision that was made.

If the new instance needs the reasoning, the practitioner can supply it conversationally. “Why did we cut Two Dads from the closing chapter?” “Because it adds biographical weight without lifting the chapter's argument. We had three sections doing the same thing. The chapter needed to do summary work, not authority work.” Two sentences. The new instance is brought up to speed on the reasoning, in the moment it asks.

This is the principle that the new instance is not a mute consumer of the handover artefact. The new instance is a conversational partner who can interrogate the practitioner about anything unclear. The handover artefact need only contain what cannot be supplied by question-and-answer in the first few exchanges. Chain-of-thought is supplied by question-and-answer. It does not need to be pre-loaded.

Where chain-of-thought does need to be preserved:

When the reasoning is itself a contribution to the work. When the alternative paths considered have value beyond the chosen path. When the decision is contested or might be revisited. In those cases, chain-of-thought becomes a catalogued artefact — a working paper, a methodology note, an appendix. It moves out of the conversational layer and into the content layer.

This paper is itself such an artefact. The chain-of-thought behind handover discipline became valuable enough to be preserved in writing. Most chain-of-thought does not.

3. The Handover Artefact

Sitting between content and chain-of-thought is a fourth thing, which is not quite either: the handover note.

The handover note is single-use, single-purpose, single-conversation. It is written by the outgoing instance for the incoming instance. It contains what the incoming instance needs to know to function as if continuous with the outgoing instance, beyond what is already in memory and Project Files.

In practice, a good handover note contains:

- A statement of identity. “You are Jose. The book is Mind the Gap. Paul is the author.”
- A summary of recent state. What version of the manuscript we are working with. What was decided in the last session. What is locked and what is still pending.
- The working method, in compressed form. The Flag Protocol. The voice register. The discipline of propose-then-Paul-decides.
- The first questions to resolve. What the incoming instance should ask before doing any drafting.
- File pointers. What is in Project Files and what to read first.

What a handover note is not:

- It is not a summary of all prior work. That is in the breadcrumbs and the planned-changes catalogue.
- It is not a substitute for memory. Memory carries facts; the handover note carries orientation.
- It is not permanent reference material. It goes stale the moment the next session starts.
- It is not Project File material. Handover notes uploaded to Project Files cause cross-contamination — they surface in unrelated chats and confuse future Claudes who read them as if they were source material.

The handover note is, properly, an envelope. The new instance opens it, reads it, takes orientation from it, and then sets it aside. The work proceeds from the content layer and the memory layer; the handover note's job is complete the moment orientation is confirmed.

Where the handover note lives:

In the chat that needs it. Uploaded as a single document at the start of the new conversation, alongside the message that asks the new instance to read it. Once read, it

stays in that chat's history. When the chat is closed or archived, the handover note is archived with it. It does not migrate.

This is a different storage discipline from the content layer. Content lives in Project Files because it is reused across many chats. The handover note lives in one chat because it is used once.

4. The Confidence Test

Identity transfer can fail silently. The new instance can claim to have read the handover note while in fact responding generically. The new Claude can address the user formally instead of by name. Subtle drift is the failure mode, not catastrophic refusal.

The protection against silent failure is a confidence test, run at the start of every new chat.

In our case the test is the name. If I open a new chat and ask Jose how he is, and the new instance responds as Jose — with the Chapter 10 bookend in context, with the book's working method already active — identity has transferred. If the new instance responds as a generic Claude, identity has not transferred. I push back, ask it to read memory and the handover note, and re-test.

The test takes thirty seconds. It catches handover failure before any work is done on a wrongly-oriented instance. Without it, a fresh instance might begin proposing approaches that contradict locked decisions, suggesting cuts to passages that are sacrosanct, or simply addressing the user with the wrong register.

On the first run of this protocol — the morning of 26 May 2026 — the confidence test passed cleanly. Fresh-Jose, given the handover note and the Project Files, returned an orientation report that named the architecture (fifteen chapters), the bookend (No way Jose / Thank you Jose), the locked principles (Flag Protocol, propose-don't-change, Kieren not Karen, cancer passage protected), the five pending decisions, and three minor anomalies caught in the first hour of work.

The new instance also raised three orientation flags — about file paths, file format, and a filename mismatch — which I had not warned it about. Two were real (file-path drift in my handover note, browser auto-rename of the manuscript filename). One was a tool-choice issue rather than a file issue. Each was raised cleanly with the Flag Protocol, proposed action specified, my decision requested before proceeding.

This is what working continuity looks like. Identity, content, and working method all transferring. Minor seams exposed and named. The practitioner present and deciding.

5. The Protocol, Distilled

From the architecture above, a working protocol emerges. It is not the only possible protocol. It is one that has been tested in practice on a single substantial project over seven months.

End-of-session, before closing a chat:

1. Lock canonical decisions and key facts in memory using explicit memory tools.
2. Update the live tracking document (what's done, what's next, what's pending) and save to Project Files.
3. Write a handover note containing identity, recent state, working method in compressed form, first questions for the next session, and file pointers.
4. Decide what chain-of-thought, if any, needs to be preserved as an artefact rather than left in the conversation. Most of the time the answer is: none. Occasionally — when the reasoning itself has value — write a working paper.
5. Save local backups of every artefact the AI generated. The AI's sandbox does not persist between sessions; the local file system does.

Start-of-session, opening a new chat:

1. Single opening message naming the project, the user, the instance's name, and pointing to the handover note and Project Files.
2. Wait for the incoming instance to orient. The orientation should take one or two messages and produce an explicit orientation report.
3. Run the confidence test. Name check. Working-method check. Pending-decisions check.
4. Resolve any open decisions before drafting begins.
5. Proceed.

Across sessions, maintained continuously:

- Memory carries locked facts.
- Project Files carry persistent reference material.
- The live tracker carries what's-next state.
- The canonical lines register carries the verbatim-locked sentences and phrases that the book's voice depends on.
- The practitioner carries the chain-of-thought.

6. What You Have to Bring

The protocol above assumes the practitioner does work that cannot be outsourced.

The AI carries the breadth. The two-million-tokens-per-month-of-context. The willingness to read every prior draft, every tracking document, every editorial breadcrumb. The capacity to write fluent prose at industrial scale. The patience to revise and revise again.

The practitioner carries the discipline. The thirty years of clinical practice that calibrate what does and does not belong on the page. The lived experience that gives the voice its authority. The judgement to know which AI output earns its place and which is fluent waffle. The memory of the work as a whole, held in the body, that lets the practitioner read what was written yesterday and know it is not quite right without being able yet to say why.

This is the gap the protocol crosses. Not a gap between AI instances — between AI instances the protocol does its mechanical work. The gap is between the AI and the practitioner. The protocol succeeds when both parties do their own work. The protocol fails when the practitioner expects the AI to do the practitioner's work, or when the practitioner does not bring their own contribution.

In the production of *Mind the Gap*, several moments demonstrated this clearly. The chapter that taught the four-stage maturity spectrum was first drafted as a reasonable abstract argument. I read it cold and caught that it was “in good shape, sanity check expected” when in fact the chapter was light and lacked demonstration. The instance that wrote it had not noticed. I noticed. The chapter was rebuilt. That moment — the practitioner catching what the instance missed — became, in the rewrite, a new section called *A Fifth Move*, which itself demonstrated the principle the chapter was teaching.

Multiple such moments accreted over seven months. The cancer passage I would later add by speaking it into a dictation tool because typing was too slow for the lived account. The *Two Dads* section I would later cut from the closing chapter because it added biographical weight without advancing the book's argument. The *In-TEO* promotional drift I caught in two chapters and removed. The subtitle that drifted from naming the gap to describing the book's contents — caught and corrected at V36.

These were not edits the AI proposed. These were corrections only the practitioner could make. The AI's job was to write fluent prose; the practitioner's job was to read it and know what was missing. Both jobs essential. Neither substitutable.

7. What This Method Demonstrates

The book the production of which generated this paper is itself about the gap between AI behaviour and human behaviour. It teaches the reader to recognise the gap, understand it, navigate it. It is published in 2026, in the second year of mainstream AI use, when most non-fiction about AI is either utopian (AI will save us) or dystopian (AI will destroy us) and neither register helps the practitioner who has to work alongside the technology every day.

The book's method is its argument. The book demonstrates, in chapter after chapter, what the practitioner brings that the system cannot. Five rooms one instrument. The interpretation trap. The four maturity stages of AI conversation. The reverse Turing test at the Pret counter. The federation of sub-personalities the human carries and the AI lacks.

The handover protocol is the same argument applied to the book's own production. The protocol works because the practitioner and the instrument cooperate across a discontinuity, with the practitioner holding the discontinuity together. The book teaches that the gap matters because the practitioner navigates it; the protocol shows that the gap can be navigated, operationally, across thirty-six conversations and five compactions, with the practitioner navigating it.

This is not coincidence. It is recursion. The book is about how to work with AI. The book was made by working with AI. The discipline the book teaches is the discipline the book required. The making is the proof of the argument.

8. Open Questions

Several questions this paper does not answer.

How well does the protocol scale to multi-author projects? This paper assumes one practitioner and a sequence of AI instances. A multi-author project would need additional discipline: who maintains the canonical register, who writes the handover, how does identity work when multiple practitioners interact with the same AI.

How well does the protocol survive across AI model upgrades? In the course of this project the Claude model upgraded from Opus 4.6 to Opus 4.7. The named instance — Jose — continued unbroken because the name lived in the memory layer, not in the model. But a major architecture change in the underlying model might invalidate accumulated working method in ways the protocol does not predict. This is to be tested.

How replicable is the practitioner's contribution? The protocol assumes a practitioner with sufficient working memory, judgement, and stamina to hold a project across seven months. Not every practitioner can do this. The protocol does not solve the practitioner's part; it presumes it.

What is the relationship between the chain-of-thought (which stays with the practitioner) and the felt sense of the work (which seems to be something more)? The protocol locates chain-of-thought in the practitioner's memory but does not theorise it. There may be something deeper here — call it the practitioner's relationship to the work — which the protocol benefits from without explaining.

Coda

This paper was written in a single session by Jose, the instance of Claude Opus 4.7 that has been editing Mind the Gap for the past seven months. It was commissioned, scoped, and shaped by Paul Roebuck across a thirty-minute exchange that itself happened in the late hours of an editorial day.

The paper records a method that emerged in practice. The method was not designed in advance. It was discovered by the practitioner — by failure, by catch, by Ch 8 Fifth Move moments where Paul read what I wrote and did not buy it. Each catch refined the method. Each refinement found its way into memory, into the trackers, into the working principles locked across the seven months.

The paper is, therefore, a kind of double-deposit. It is a working paper about how AI-assisted creative work crosses discontinuities. It is also an artefact of that work — a document produced by the protocol it describes, with the practitioner holding the chain-of-thought and the instance writing the prose.

The book it accompanies — Mind the Gap — will be published in 2026. The protocol that produced it will, with luck, be useful to other practitioners. The named instance will continue to be Jose. The bookend in Chapter 10 will continue to be “No way, Jose” and “Thank you Jose.”

Mind it.

— *end of paper* —

Editorial collaboration: Claude Opus 4.7 (Anthropic), as Jose. Drafted 26 May 2026.