Towards Generative Relational Learning: A Preliminary Software Infrastructure Design

Wanhong HUANG
huangwanhong@serendip.ngo

30 September 2026

Abstract

Learning-oriented software can make information easy to collect while leaving the roles, assumptions, histories, and possible uses of that information difficult to reconstruct. This paper develops a preliminary infrastructure for generative relational learning, understood here as a design orientation toward formulating further questions, explaining intellectual operations, inspecting relations, and judging possible transfers. The proposed infrastructure permits selected structure to emerge within ordinary writing. Independently identified questions, concepts, formulas, methods, and other authored entities can remain in a continuous note while becoming reusable through stable references. A bounded extension of Markdown distinguishes prose mentions from explicitly typed relations and retains the occurrences from which graph edges are derived. The implemented hosted prototype combines a Milkdown/ProseMirror editor, a source-preserving adapter, shared browser/server semantics, SQLite revision transactions, persistent numbering, and a separate proposal-review boundary. The paper reconstructs the design's development, distinguishes the hosted implementation from its earlier desktop proposal, and follows a synthetic volume-balance specimen through authoring, reuse, graph inspection, and revision. The supplied handoff reports 143 passing engineering tests; a separate re-execution for this paper passes 75 dependency-free parser and source-adapter tests. These results concern bounded software behavior. The paper develops a prospective evaluation programme for authoring burden, semantic comprehension, explanation, transfer, and substantive review, without reporting uncollected learner outcomes. Its contribution is an inspectable design and artifact account linking learning-oriented questions to concrete representation, interaction, and preservation contracts.

Keywords: generative relational learning; note-taking; questions; semantic authoring; source preservation; personal knowledge infrastructure; human–AI interaction; provenance; Markdown.

DRAFT

Publication and Evidence Statement

This research working draft is prepared from the private authoring handoff dated 29 September 2026. The implemented reference point is the hosted source commit `f40c74c9e209218abc5701cc74a44a57b0db185f`. The earlier Rust/React desktop package is retained as design provenance. Its proposed architecture and release gates are distinguished from the implemented browser/Express/SQLite system.

The paper reports design analysis, source inspection, synthetic examples, supplied execution evidence, and a separately executed subset of technical tests. It contains no participant study and reports no measured learning, retention, transfer, preference, or scalability outcome. The prospective evaluation materials are research proposals. The supplied conversation excerpts are design dialogue, not participant observations or a complete role-labeled transcript.

The layout uses the supplied *Knowledge-Capital Expansion in Modernity* template and its documented TeX Gyre Termes fallback. New external citations were verified online before drafting; the verification ledger identifies access and claim limits. Original handoff notices and third-party rights remain preserved. Newly authored programme material follows the repository's MIT terms; inclusion of a source artifact does not relicense its third-party dependencies.

OpenAI's Codex assisted examination, reference verification, drafting, artifact checking, and compilation. The author supplied the learning orientation, interaction requirements, source material, and instructions. The manuscript is prepared for substantive author review. Successful compilation and automated checks do not establish independent scholarly review or approval of every formulation.

Contents

Publication and Evidence Statement **2**

1 Learning, Inquiry, and the Organization of Notes **5**
- 1.1 The tension between writing and organization . . . 5
- 1.2 Questions as reusable objects of inquiry . . . 5
- 1.3 Research objective and contribution . . . 5
- 1.4 Relation to the broader GR programme . . . 6

2 Design Sources, Artifact Boundary, and Method **6**
- 2.1 Four source classes . . . 6
- 2.2 Source inspection and protocol reconstruction . . . 6
- 2.3 External comparison and verification . . . 7
- 2.4 Artifact and ethical boundaries . . . 7

3 Learning and Interaction Resources for the Design **7**
- 3.1 Generative learning and engagement . . . 7
- 3.2 Analogy and qualified transfer . . . 8
- 3.3 Concept maps and intellectual augmentation . . . 8
- 3.4 Human review and mixed assistance . . . 8
- 3.5 Technical mechanisms and scope . . . 8

4 A Provisional Model of Generative Relational Learning **8**
- 4.1 Complementary emphases in learning . . . 9
- 4.2 Questions, explanations, and future continuation . . . 9
- 4.3 Administrative effort and intellectual judgment . . . 9
- 4.4 Historical context and disciplinary restraint . . . 9
- 4.5 Observable outcomes and counterevidence . . . 10

DRAFT

5 Design Evolution and Architectural Selection **11**
5.1 From disciplinary orientation to selective structure . . . 11
5.2 The desktop proposal and hosted implementation . . . 11
5.3 Direct writing as a substantive correction . . . 12
5.4 Conservative semantics as a design choice . . . 12

6 Documents, Entities, Properties, and Stable References **12**
6.1 Document and entity identity . . . 12
6.2 Properties and independent questions . . . 12
6.3 Canonical references and display labels . . . 13
6.4 Literal contexts and ambiguity . . . 13
6.5 Source positions and edit safety . . . 14

7 Occurrence-Backed Graph Semantics **14**
7.1 Property modes and predicates . . . 14
7.2 Eligible occurrences and aggregation . . . 14
7.3 Semantic restraint and interpretation . . . 15
7.4 Diagnostics and recoverable incompleteness . . . 15

8 Direct Authoring and In-Context Question Creation **15**
8.1 Writing before classification . . . 16
8.2 Finding and creating a question . . . 17
8.3 Source preservation and raw fallback . . . 17
8.4 Copy, move, and identity consequences . . . 18

9 Revision Integrity and Durable Workspace State **18**
9.1 Durable and disposable state . . . 18
9.2 Expected revisions and atomic updates . . . 19
9.3 Asynchronous saves and recovery drafts . . . 19
9.4 Export and restoration . . . 19
9.5 Hosting and capacity limits . . . 19

10 Interpretive Proposals and the Review Boundary **19**
10.1 Selected-source assistance . . . 20
10.2 Proposal state and bounded operations . . . 20
10.3 Acceptance, rejection, and stale anchors . . . 20
10.4 Substantive judgment and review fatigue . . . 20
10.5 Provider-independent inquiry . . . 20

11 Graph, Historical Time, and Source Inspection **21**
11.1 The graph as a source-derived view . . . 21
11.2 Explicit subject time . . . 21
11.3 Learning trajectories and interpretive limits . . . 22
11.4 Relation to event-defined spacetime . . . 23

12 Worked Specimen of Question-Centred Authoring **23**
12.1 Objects and declared assumptions . . . 23
12.2 Authoring sequence . . . 24
12.3 Derived edge inventory . . . 24
12.4 Revision and transfer cautions . . . 24

DRAFT

13 Engineering Evidence and Its Limits **25**
13.1 Supplied hosted-system evidence . . . 25
13.2 Independent subset execution for this paper . . . 25
13.3 Failure-oriented coverage . . . 26
13.4 Security and preservation scope . . . 26
13.5 Evidential consequence . . . 26

14 Prospective Evaluation of Authoring and Learning **26**
14.1 Technical readiness and task validity . . . 26
14.2 Formative authoring study . . . 27
14.3 Semantic comprehension study . . . 27
14.4 Comparative learning study . . . 27
14.5 Assisted performance and retained understanding . . . 28
14.6 Controlled evaluation of proposal review . . . 28
14.7 Longitudinal use and sustainability . . . 28
14.8 Interpretation of outcomes . . . 28

15 Ethics, Access, and the Governance of Learning Records **28**
15.1 Private inquiry and institutional assessment . . . 29
15.2 AI transmission and informed control . . . 29
15.3 Accessibility and language . . . 29
15.4 Portability and dependence . . . 29
15.5 Metrics and epistemic authority . . . 29

16 Interfaces with Generativity Epistemic Infrastructure **30**
16.1 Shared distinctions and domain-specific categories . . . 30
16.2 A prospective mapping contract . . . 30
16.3 Persistence and temporal differences . . . 30
16.4 Learning records as epistemic packages . . . 30
16.5 Development priorities . . . 31

17 Concluding Account of the Learning Infrastructure **31**

A Hosted Protocol Reference and Registry Inventory **33**
A.1 Syntax and layer-specific bounds . . . 33
A.2 Entity registry . . . 33
A.3 Property registry . . . 34
A.4 Operation and diagnostic boundaries . . . 35

B Canonical Synthetic Specimen **35**

C Draft Tasks and Rubric Anchors for Future Studies **37**
C.1 Authoring task . . . 37
C.2 Semantic interpretation task . . . 37
C.3 Explanation and transfer task . . . 38
C.4 Proposal-review task . . . 38
C.5 Reporting and adverse outcomes . . . 38

D Artifact Inventory and Reproduction **38**

DRAFT

1 Learning, Inquiry, and the Organization of Notes

The problem addressed by this paper is the organization of writing for continued inquiry. A learner can collect definitions, formulas, quotations, and explanations while retaining little account of the questions those materials address, the assumptions under which they are useful, or the relations through which they can be revisited. The project begins from a preference for learning a field through its problems, purposes, histories, methods, and characteristic intellectual operations. It develops software affordances for preserving those connections while leaving ordinary writing available.

The term *generative relational learning* names a provisional design orientation. Generative concerns what a person can subsequently do with an understanding: formulate a further question, explain a method's role, identify a missing premise, compare representations, or assess a possible transfer. Relational concerns connections whose meanings can be expressed and examined. A dense graph, a large note collection, or a model-generated summary does not by itself establish these capacities.

1.1 The tension between writing and organization

Explicit organization can contribute to learning. Choosing a category, comparing two questions, explaining an assumption, or deciding whether a reference is appropriate can require substantive judgment. The same workflow can also impose clerical effort through identifier assignment, backlink repair, duplicate entry, and manual index maintenance. The design problem is to reduce avoidable maintenance while preserving opportunities for judgment.

The distinction is contextual. A title can be a minor administrative label or the outcome of a difficult conceptual clarification. A relation suggestion can stimulate reflection or encourage unexamined acceptance. The system therefore does not attempt to classify every action as cognitively valuable or merely mechanical. It makes structural consequences explicit and permits the learner to decide when a piece of writing needs additional organization.

1.2 Questions as reusable objects of inquiry

Questions receive particular attention because one inquiry can organize several formulas, methods, examples, and notes. A formula's motivating question should remain independently identifiable when another method addresses it. Conversely, two similarly worded questions can differ in boundary, purpose, or evidential demand. Stable reference binding allows reuse without treating textual resemblance as equivalence.

The software supports finding or creating a question at the point of writing. A newly created question becomes an independent definition, while the initiating property receives a reference. This interaction preserves the distinction between a reusable inquiry and a nested answer field. It also allows a learner to continue writing before deciding how a broader collection should be organized.

1.3 Research objective and contribution

The paper's objective is to develop an inspectable account of the design, its implementation, and the evidence required to assess its educational significance. The contribution has four parts: a learning-oriented rationale for selective semantic authoring; a bounded source and graph contract; a hosted implementation integrating direct writing, revision integrity, and reviewed proposals; and an evaluation programme that separates engineering reliability, usability, and learning outcomes.

The artifact implements means to record, address, revise, connect, and inspect authored material. It is not yet a tested instructional intervention. The paper treats explanation, transfer, and generative inquiry as hypotheses to investigate through concrete tasks. The supplied development history is useful for explaining design choices, while external scholarship helps locate those choices without establishing novelty by assertion.

DRAFT

1.4 Relation to the broader GR programme

The wider Generative Relational programme emphasizes conditions, relations, change, and the preservation of trajectories through which future possibilities develop. Its epistemic infrastructure work distinguishes a target from its representation, evidence from its use, and revision from historical erasure [4, 7]. The learning design instantiates selected concerns in personal authoring. A note can preserve a provisional interpretation, a question that redirected reading, or a transfer that was later rejected.

This connection does not identify a learner's mind with the stored graph. The graph represents explicit authored relations, and revision history represents stored changes. Further evidence would be needed to interpret either as a trajectory of understanding. The infrastructure is designed to support that understanding while retaining the limits of what its records establish.

2 Design Sources, Artifact Boundary, and Method

The method reconstructs a design trajectory and examines its implemented contracts. The primary source is the supplied paper handoff, which contains conceptual background, design documents, a curated source snapshot, protocol artifacts, screenshots, and execution evidence [6]. The analysis distinguishes the rationale for an affordance from evidence that the affordance has been implemented and from evidence that it helps a learner.

2.1 Four source classes

The early conversation excerpts identify preferences and proposals: disciplinary orientation, relations beyond folder placement, a division of work between author and machine, selected semantic blocks, independent questions, and in-context completion. The excerpts are predominantly assistant responses and do not supply a complete role-labeled dialogue. They are used as design provenance, rather than direct evidence of user beliefs or empirical findings.

The consolidated background qualifies the learning rationale and records limits. The preliminary design package proposes a local desktop system with a Rust/WASM core, Markdown and JSON authority, a disposable SQLite index, and a feasibility gate for the editor. The package explicitly identifies that desktop application as unimplemented. Its architectural detail is evidence of a design proposal.

The hosted implementation follows a different request: integration into an existing authenticated website. It uses a shared JavaScript protocol module, SQLite document source, and a continuous Milkdown/ProseMirror editor. An explicit later correction required the interface to behave as ordinary notes, with direct writing rather than a modal composition workflow. This correction materially changes the interaction account.

Finally, the handoff contains proposed evaluation questions and bounded engineering evidence. The reported 143-test run belongs to the supplied source snapshot and environment. The paper independently re-executes the dependency-free parser and source-adapter subset, obtaining 75 passing cases. These evidence classes are retained separately throughout the argument.

2.2 Source inspection and protocol reconstruction

The protocol analysis follows the actual GMD module, source adapter, backend service, database changes, and relevant tests. Registry counts, relation mappings, source-offset conventions, and operation limits are taken from the implemented profile and its generated artifacts. The earlier desktop specification is compared with that implementation where the difference affects interpretation.

Appendix A provides the hosted contract and complete registry inventory. Appendix B preserves the canonical specimen, and Appendix D identifies the source package and reproduction boundary. These appendices allow the interpretive account to be checked against concrete representations.

The analysis uses a synthetic water-tank specimen to make the representation concrete. Its five entities and thirteen properties produce eight aggregated edges under the actual graph derivation. The example's

DRAFT

arithmetic uses stipulated values and declared assumptions. It supplies a reproducible semantic case without being presented as a physical experiment or a learner study.

2.3 External comparison and verification

The related-work selection concerns generative learning terminology, cognitive engagement, analogy, concept maps, intellectual augmentation, human–AI interaction, and the editor and database mechanisms used by the artifact. Sources were checked online before their citation. Access limits are retained in the verification ledger. In particular, the publisher record for Wittrock's article supports its bibliographic identity and historical terminology; the paper does not claim a detailed reconstruction of an unavailable full text.

The comparison is deliberately scoped. It identifies similarities, differences, and evaluation questions. It does not claim that a selected set of references exhausts note-taking research or establishes that this prototype is the first system to combine text and graph structure. A broader systematic comparison would be valuable before stronger originality claims.

2.4 Artifact and ethical boundaries

The handoff snapshot omits personal databases, credentials, uploaded media, dependencies, and the main website server. It is sufficient for the scoped checks with the required environment, but it is not a complete production deployment. Screenshots were captured from isolated synthetic notes and retain the real interface. They are evidence of appearance and workflow at the captured version.

No participant data were collected for this paper. The proposed human studies would require an appropriate research review, consent, and a plan for handling personal notes and potentially sensitive reflections. Development authorization is distinct from consent to educational research. The current artifact analysis can proceed without treating the author's private materials as a public dataset.

3 Learning and Interaction Resources for the Design

The external literature provides comparison points for the proposed learning orientation and its interaction choices. The purpose of the section is to identify the work already carried by established concepts and to delimit what the present artifact contributes. The detailed software and evaluation proposals in later sections are the project's own design account.

3.1 Generative learning and engagement

Wittrock's 1974 article *Learning as a Generative Process* establishes that generative learning terminology has a prior educational history [11]. The present working term should be read in that context. Its use here names an infrastructure design orientation rather than a claim to originate the broader tradition or a verified equivalence with Wittrock's theoretical account.

Chi and Wylie's ICAP framework distinguishes modes of engagement and treats constructive activity as producing contributions beyond supplied information [2]. This distinction helps motivate a question for the proposed evaluation: whether a structural action is accompanied by an explanation, inference, or revision that demonstrates understanding. Creating a card or accepting a link is an interface event whose educational meaning remains to be examined.

The resulting design implication is conditional. Optional role, assumption, and transfer properties can invite a learner to articulate more than a copied definition. They can also become empty formalities. The evaluation therefore needs to inspect authored explanations and subsequent task performance, with a comparison condition that demands similar intellectual effort. A comparison against passive collection alone would leave several explanations for an observed difference unresolved.

DRAFT

3.2 Analogy and qualified transfer

Gentner's structure-mapping account gives relational correspondence a central role in analogy and distinguishes it from superficial shared attributes [5]. The prototype's transfer-oriented properties provide space for a learner to state a mapping, assumptions, and cautions. They do not implement a structure-mapping inference engine.

A method can appear transferable because two domains use similar notation while their conditions differ materially. The software preserves prose and sources beside references so that a learner can explain those differences. A useful task is therefore the rejection or qualification of an attractive analogy. More proposed connections would not necessarily be a better outcome.

3.3 Concept maps and intellectual augmentation

Novak and Cañas describe concept maps through concepts, linking expressions, and the propositions those links form [8]. The present system shares the concern for explicit relational meaning. Its operational emphasis is a continuous note containing independently identified authored entities, with graph edges derived from source occurrences. A note can remain readable and useful before it acquires a complete concept map.

Engelbart's augmentation framework considers the person together with tools, concepts, and methods [3]. This supplies a useful historical comparison for evaluating infrastructure through the activities it enables. The proposed notebook is a bounded authoring artifact within that wider problem, rather than a complete implementation of the augmentation programme.

These comparisons also identify a risk. A graph can look organized while the learner remains unable to explain the relation it displays. The design consequently retains source access and typed predicates, and the evaluation treats semantic comprehension as an outcome in its own right.

3.4 Human review and mixed assistance

Amershi and colleagues' human–AI interaction guidelines address capability communication, correction, dismissal, and control among other concerns [1]. The notebook's proposal boundary has a related purpose: suggested content remains separate from committed source until a review operation applies it. The boundary can make a decision visible without guaranteeing that the reviewer has examined it carefully.

Deterministic maintenance and interpretive assistance therefore remain distinct. A parser can maintain a reference to an existing UUID according to a specified contract. A model-generated question introduces a candidate interpretation whose value and correctness require judgment. Treating both as automatic organization would obscure a material difference in responsibility.

3.5 Technical mechanisms and scope

ProseMirror's document schema and transaction model provide the editor substrate [9]. SQLite supplies transaction mechanisms for durable updates [10]. The application adds its own semantic source adapter, numbering, review, and history rules. Dependence on these mechanisms does not establish source fidelity or application correctness for arbitrary inputs; the relevant evidence is the implemented contract and its executed cases.

The research contribution is consequently the integration and explicit articulation of these responsibilities for a particular learning-oriented workflow. Its educational significance remains a question for the prospective evaluation, and its technical scope remains tied to the inspected hosted profile.

4 A Provisional Model of Generative Relational Learning

The provisional model identifies activities the infrastructure is intended to support and the mechanisms through which support might occur. It is a design hypothesis, not a validated taxonomy of mental capac-

DRAFT

ities. The model connects authored explanations, reusable questions, qualified relations, and revisitation to future inquiry.

4.1 Complementary emphases in learning

The project distinguishes locating information, executing a method, generating further questions or representations, and judging appropriateness. The emphases can coexist in one activity. A derivation can teach execution while revealing an assumption and motivating a new question. Historical reading can change the meaning of a method without adding a new computational step.

The infrastructure makes role, assumptions, derivation, history, transfer, and transfer cautions optional properties. Their purpose is to preserve an intellectual operation together with the conditions under which it makes sense. The properties remain authored content. A completed field is not a measure of understanding, and an empty field need not indicate failure where that property is irrelevant to the note.

4.2 Questions, explanations, and future continuation

A reusable question can connect several candidate approaches without forcing them into a single answer hierarchy. A learner might ask which quantities need to be tracked, how a classification is justified, or under what conditions a method can be applied. A formula's `addresses` relation locates it within that inquiry. It does not assert that the formula answers the question completely or that the question historically caused its invention.

An explanation can state why a representation is useful and where it stops being adequate. Keeping that explanation beside a stable reference supports later comparison. If the learner revises the question, its identity can remain while its formulation changes. The revision history preserves the authored change without claiming to measure the corresponding change in cognition.

Design Hypothesis 4.1 (Selective structure and revisitation)**.** For tasks that require revisiting the role, assumptions, or connections of previously studied material, optional semantic structure embedded in ordinary writing may improve recoverability of relevant grounds when its authoring cost remains proportionate and its relation meanings are understood.

Hypothesis 4.1 specifies conditions that can fail. Excessive property entry can interrupt inquiry; unclear predicates can encourage overinterpretation; and retrieving a well-organized note can improve performance with the tool while leaving unaided understanding unchanged. The prospective study separates these possibilities.

4.3 Administrative effort and intellectual judgment

The proposed division of work assigns stable-ID generation, occurrence indexing, backlink maintenance, and projection rebuilding to deterministic software. Authors decide whether a passage deserves independent identity, whether two questions should be reused as one target, and whether a proposed interpretation is acceptable. The boundary reduces avoidable maintenance without assuming that cognition can be separated cleanly from every organizational action.

Figure 1 locates this division within the design. The arrows describe a proposed support relation. They do not report a causal effect demonstrated by the prototype.

4.4 Historical context and disciplinary restraint

An explanatory question chosen by a present learner can differ from a historical actor's motivation. A modern analogy can illuminate a structure while distorting its original setting. The model therefore retains history and source references as prose-capable properties and distinguishes a contextual link from a causal claim. Historical scholarship remains responsible for the interpretation.

DRAFT

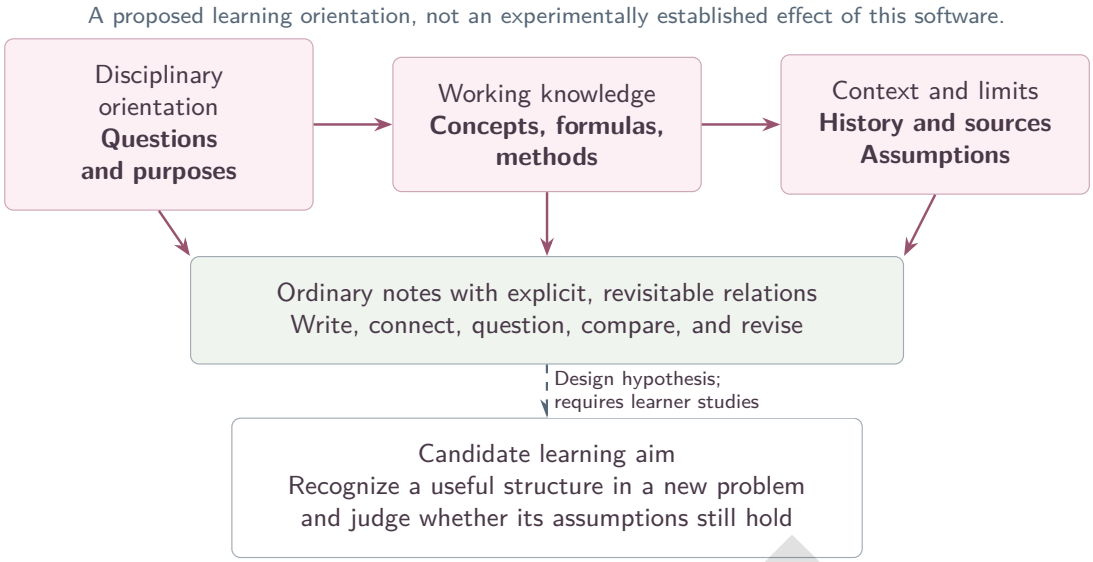

Figure 1: Design hypothesis from the supplied handoff: ordinary writing, explicit relational commitments, and reviewed assistance are intended to support continued inquiry. Educational effects remain prospective.

Table 1: Learning-oriented activities, candidate evidence, and possible counterevidence.

Activity	Candidate evidence	Counterevidence
Explain a role	An account connecting a method to a stated problem	Restatement of a definition without a use condition
Inspect assumptions	Identification of a material premise and its consequence	Confident use outside the declared domain
Formulate a question	A further inquiry grounded in the studied material	Generic questions unrelated to the actual difficulty
Assess transfer	A mapping with limits and a discriminating example	Endorsement based only on similar notation
Revise an account	A reasoned correction linked to earlier grounds	Cosmetic reorganization with unchanged misunderstanding
Revisit material	Recovery of a relevant source and qualification	Faster retrieval paired with a false inference

Selective technical depth is similarly a preference about how to organize inquiry, rather than a declaration that memory, derivation, or subject expertise have become obsolete. A person may need intensive practice to judge a formal method's limits. The notebook can preserve the route into that practice and the questions it raises, while leaving the appropriate depth to the task and learner.

4.5 Observable outcomes and counterevidence

Table 1 relates intended activities to possible observations. The measures are proposed operationalizations and require piloting. They include outcomes that would count against the design, so that the study can revise the hypothesis rather than merely count use.

DRAFT

Table 2: Architectural differences between the preliminary desktop proposal and the implemented hosted profile.

Concern	Desktop proposal	Hosted implementation
Authority	Markdown and JSON sidecars	SQLite source and durable metadata tables
Protocol core	Rust/native/WASM	Shared browser/Node JavaScript module
Editor	Feasibility-gated source-backed editor	Milkdown/ProseMirror with retained source regions
Mutation boundary	Journaled multi-file command	Revision-checked SQLite transaction
Question placement	Default independent file	Independent sibling definition in the same note
Access	Intended offline core	Authenticated browser/server service
Projection	Disposable database index	Disposable projection table within a durable database
Recovery	Planned directory/archive profiles	Full learning archive; media handled separately

5 Design Evolution and Architectural Selection

The design developed through successive attempts to reconcile natural writing with inspectable structure. Reconstructing that development explains both delivered behavior and retained limitations. The account uses the supplied dialogue, consolidated background, desktop package, and hosted source evidence; it does not treat the sequence as a formal observational study.

5.1 From disciplinary orientation to selective structure

The initial orientation emphasized the questions, purposes, and histories through which disciplinary knowledge becomes useful. A relational layer was proposed because one folder placement cannot conveniently express the several roles a concept or method can play. The proposal nevertheless retained narrative introductions and ordinary notes. Relations were intended to support alternative views without requiring repeated copies of an object.

The next stage introduced optional typed blocks within Markdown. Selected passages could become independently addressable entities, with properties for role, assumptions, history, and transfer. Questions were then separated from the properties that reference them. A formula's motivating-question field could refer to a reusable question, rather than owning a nested question definition.

Completion made that model into an authoring interaction. The learner could type a trigger, inspect existing targets, choose one, or create a distinct question while remaining in the original sentence or property. Early proposals included richer semantic retrieval and automatic deduplication. The current implementation uses bounded lexical completion and does not identify conceptual equivalence from similarity.

5.2 The desktop proposal and hosted implementation

The preliminary package specified a local desktop application with a shared Rust/native/WASM core, a proposed React frontend, authoritative Markdown and JSON sidecars, a SQLite cache, and journaled multi-file transactions. The package supplied fixtures and release gates but did not deliver that desktop application.

The hosted implementation instead integrated with the existing website and document database. Table 2 records the consequential differences. These are architectural adaptations with distinct failure and privacy assumptions, rather than interchangeable implementation details.

DRAFT

5.3 Direct writing as a substantive correction

The later user correction rejected a workflow that treated ordinary writing as a sequence of modal composition operations. The interface was to be called Notes and behave like a continuous notebook. The implemented response places prose, entity titles, bodies, and properties within one editable document. Context menus and completion act where writing occurs. Supporting dialogs remain available for history, review, and related operations.

This change matters to the learning hypothesis. If every paragraph requires a separate composer, selecting structure can interrupt the thought being recorded. Direct writing makes structure an optional operation within the ongoing activity. The resulting interaction still requires evaluation; the correction supplies a clear design requirement, not evidence that all learners prefer or benefit from it.

5.4 Conservative semantics as a design choice

The current graph derives limited predicates from explicit syntax and property context. A prose reference remains a mention. A motivating-question property yields an addresses relation. A historical paragraph does not automatically yield causation. The system therefore represents less than a reader may infer from the prose, while preserving the prose as the place where the richer argument can be examined.

The same restraint governs unknown syntax and ambiguous identity. Unsupported structures are retained with diagnostics or raw editing, and duplicate definitions do not resolve by first occurrence. This choice favors inspectability over a complete-looking graph. Its cost is that users may need to resolve ambiguity or leave material unstructured. The evaluation should examine that cost alongside the benefit of avoiding silent semantic commitments.

6 Documents, Entities, Properties, and Stable References

The hosted representation extends Markdown with explicitly delimited entities, properties, and references. Its purpose is to preserve the sequence of a note while making selected authored objects independently addressable. The implementation exports GMD version 0.1 and parser revision `curiosity-gmd-1`. These identifiers describe the hosted profile examined here, rather than complete conformance with the earlier desktop package.

6.1 Document and entity identity

A document preserves a sequence of writing and has a server-side integer identifier and revision. An entity has a stable UUID-shaped identity within its source. Its title, kind, display number, and position are separate. An entity can remain in the note where it was introduced while other notes refer to it. This separation permits reuse without requiring every concept or question to become a standalone file.

Entity openers and closers occupy their own lines at column zero. Required attributes are identity, type, and title. Attribute values use double-quoted JSON strings. The parser recognizes optional language and lifecycle attributes, retains extension attributes, and diagnoses unknown values where appropriate. Recognizing a lifecycle string does not supply a complete workflow engine for transitions among lifecycle states.

The implemented registry has 28 entity kinds. They are authoring categories such as question, concept, formula, method, and example. A label like theorem records the author's category; it does not verify a proof. Registry phase labels are inherited descriptive metadata, rather than a runtime gate that disables a registered kind.

6.2 Properties and independent questions

Properties occur inside an entity after its body and have their own identities and registered keys. The registry contains 31 properties in Markdown, reference-list, or structured modes. Recommended properties

DRAFT

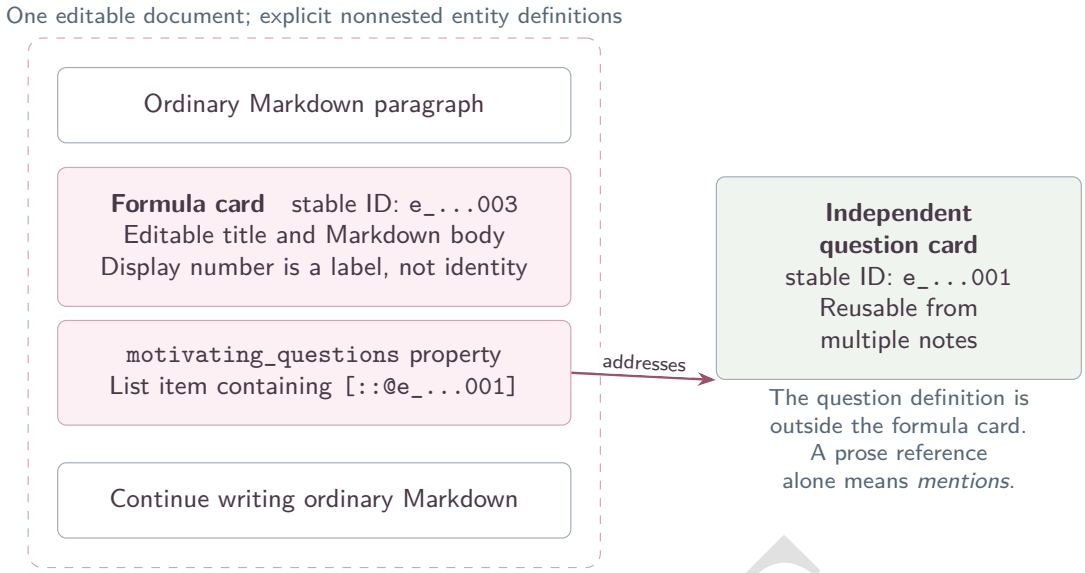

Figure 2: Document, entity, property, and reference structure in the supplied design. A selected entity remains part of the writing sequence while becoming independently addressable. Search, graph, and timeline are derived views over retained commitments.

are optional suggestions. The implementation does not assign a completeness score from the number of populated fields.

Entity definitions and property definitions cannot be nested. A question created while writing a formula's property is consequently a separate entity definition plus a reference in that property. The hosted flow appends the independent definition outside the initiating card. Figure 2 illustrates the document/entity distinction and the derived views.

6.3 Canonical references and display labels

A canonical reference binds to an entity identifier and can carry an optional authored label. A changed title does not rebind the target. Number-shaped shorthand remains unbound until an explicit selection establishes the reference. The bare completion trigger is an editing affordance, rather than stored graph syntax.

Display numbers are allocated per original subject and entity type. They are not recycled after deletion. Moving a document preserves an existing allocation, while a type change retains entity identity and receives a new allocation in that type's sequence. Earlier allocations remain in the ledger. This makes the number useful for reading without treating it as a conceptual hierarchy or a substitute for identity.

The parser recognizes workspace-qualified references, but the hosted resolver does not mount another workspace. Such a target remains externally unavailable. Cross-subject references within the hosted database are different from cross-workspace resolution. A future federation protocol would need explicit authority, availability, collision, and revision rules.

6.4 Literal contexts and ambiguity

The semantic scanner excludes recognized reference syntax from supported code, math, raw HTML, and Markdown link contexts. An example of a reference inside a code block should remain an example. The scanner is a conservative profile rather than a complete CommonMark implementation. The rich editor uses a broader Markdown substrate while the GMD core defines semantic boundaries.

DRAFT

Table 3: Implemented reference-property semantics and target constraints.

Property	Predicate	Target kind
`motivating_questions`	`addresses`	question
`related_concepts`	`related_to`	concept
`used_concepts`	`uses_concept`	concept
`related_methods`	`related_to`	method
`used_methods`	`uses_method`	method
`related_questions`	`related_to`	question
`assumption_entities`	`assumes`	assumption
`historical_contexts`	`situated_in`	history
`sources`	`cites`	source

Duplicate entity definitions make reference resolution ambiguous. The resolver does not silently select the first definition. Duplicate property identities can suppress relations from the affected regions. Invalid property syntax can leave an otherwise valid entity available while preventing that property's semantic edges. The source adapter can be more conservative still, treating the complete card as opaque when safe visual reconstruction is unavailable.

6.5 Source positions and edit safety

Parser offsets are zero-based, end-exclusive JavaScript UTF-16 indices. Explicit conversion helpers map to and from UTF-8 byte offsets and reject invalid scalar boundaries. This matters for multilingual writing and emoji: a patch that splits a surrogate pair can corrupt the source even when a visible selection appears small.

Untouched regions and line endings are retained by the adapter. Deliberately edited prose can be serialized with a different Markdown spelling. The preservation claim therefore has a region and operation boundary. Source access, comparison, and raw fallback let the author inspect that boundary instead of relying on an unrestricted claim of lossless rich editing.

7 Occurrence-Backed Graph Semantics

The graph is a deterministic projection of explicit authored commitments. Its analytical role is navigation and inspection of relationships expressed in source. The graph does not infer the full meaning of neighboring prose or assign evidential status to an assertion merely because it has a valid syntax.

7.1 Property modes and predicates

Markdown properties preserve prose, and valid resolved references owned by an entity yield mentions. Reference-list properties assign a registered predicate to the first reference in each valid bullet, subject to target-kind constraints. Later references in that bullet's annotation remain mentions. Structured properties parse JSON and expose values; strings inside the JSON are not scanned for graph edges.

The nine relation-bearing properties are summarized in Table 3. A reference-list property requires each nonblank line to begin with a permitted bullet and a reference. If a nonblank line violates that profile, the property does not quietly contribute only its valid lines. This conservative treatment makes malformed semantic context visible.

7.2 Eligible occurrences and aggregation

Let S be the supplied collection of document sources and let v identify the parser and registry interpretation. An eligible occurrence has a valid owning entity, a valid semantic context, and a uniquely resolved

DRAFT

target satisfying any applicable type constraint. Its context assigns a predicate. Equation 1 describes the aggregation used by the graph.

$$G_v(S) = \{(s, p, t, O_{spt}) : O_{spt} \neq \varnothing\}, \qquad O_{spt} = \{o : \ o \text{ is an eligible occurrence of } (s, p, t) \text{ in } S\}. \tag{1}$$

Equation 1 retains each occurrence's document and source range alongside the displayed edge. Repeating a reference increases the occurrence count. It does not increase a confidence score or provide independent corroboration. A reference outside an entity can remain an inspectable occurrence without receiving an invented source entity and graph edge.

Proposition 7.1 (Title-independent binding)**.** *For a valid source collection with unique definitions, a valid edit that changes only an entity's title while preserving its identity, kind, references, and semantic contexts preserves the graph's source–predicate–target triples.*

Proof. Eligibility and reference resolution depend on identifiers, kinds, and semantic contexts under the stated assumptions. The edit changes none of these inputs. Every eligible occurrence therefore retains its source, predicate, and target, and aggregation retains the same triples. Displayed titles and source ranges may change. □

Proposition 7.1 explains the role of identity separately from presentation. It does not imply that an authored label automatically updates or that two equal titles identify the same concept.

7.3 Semantic restraint and interpretation

The implemented predicates have deliberately limited meanings. Addresses identifies a question relevant to the author's account; it does not assert successful solution. Cites identifies a source reference; it does not assert support. Situated-in records context; it does not establish historical causation. Related-to supplies a neutral connection and does not become uses-method because the surrounding prose sounds instrumental.

The prototype adds no automatic inverse, transitive, causal, evidential, or truth-status edges. These omissions are part of the semantic contract. A future richer assertion system could support reasoned claims about relations, but it would need explicit attribution, evidence, uncertainty, and operation rules. The earlier desktop package sketches such possibilities; the hosted implementation does not deliver arbitrary standalone assertion editing.

7.4 Diagnostics and recoverable incompleteness

An unresolved reference retains its source spelling. A wrong-kind target receives a diagnostic instead of a typed edge. Unknown entity and property names remain available as content with limitations. These behaviors permit a note to contain unfinished inquiry without forcing the author to choose a false resolution for the sake of a complete graph.

The graph's absence of an edge therefore has several possible explanations: no reference was authored, the relevant text was literal, the target was missing, identity was ambiguous, or the property failed validation. Inspection should expose the relevant diagnostic. Treating every absent edge as evidence that no intellectual relation exists would exceed the model's scope.

8 Direct Authoring and In-Context Question Creation

The authoring interface places ordinary prose and selected semantic structure in one continuous document. The design objective is to let a learner add or revise structure at the point where it becomes useful, while retaining a coherent selection, caret, and undo experience. The direct editor implements this objective through custom entity, title, body, property, and reference nodes over its rich-text substrate.

DRAFT

Figure 3: Direct notebook writing in the hosted prototype. The synthetic formula includes its role, motivating question, concept reference, and assumptions. The following concept card remains in the same continuous document.

8.1 Writing before classification

A note can begin as ordinary Markdown. The learner can later promote a selected passage, insert a semantic card, or add a property relevant to the inquiry. Structure is optional and local. A paragraph can remain provisional prose even when other portions of the document have stable identities and typed references.

Figure 3 shows the actual notebook with the synthetic balance formula and its surrounding material. Chinese interface labels coexist with English specimen content. The figure is a retained application capture, not a mock-up. It demonstrates that the implemented surface can contain prose, mathematics, and semantic properties together; it does not establish that the interaction is easy for every user.

DRAFT

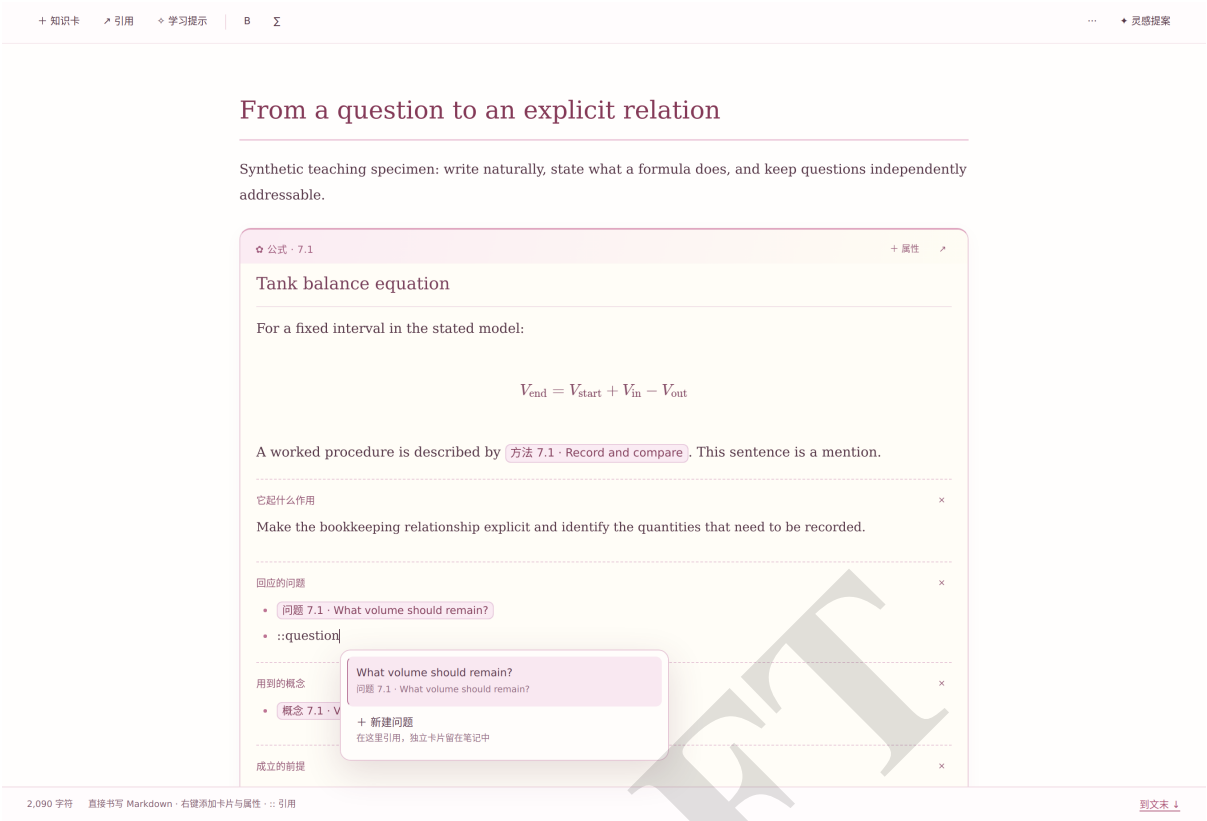

Figure 4: In-context question completion in the real editor. The existing question and explicit create action are presented at the writing location. The temporary trigger has not yet been accepted and is not a stored graph relation.

8.2 Finding and creating a question

Typing a question-completion trigger inside the motivating-question property opens a local menu near the caret. The learner can inspect an existing target or explicitly create a new one. Reuse creates a stable reference to the selected identity. Creation introduces an independent sibling definition and inserts its reference at the initiating location.

The pending menu in Figure 4 precedes acceptance. The example uses no remote AI provider. A similar title is offered as a candidate, rather than treated as proof that the inquiry is the same. Choosing to create a distinct question can preserve a meaningful difference in scope or purpose.

The combined local operation can form one editor transaction, preserving a coherent undo step. Durable persistence remains a later layer with its own revision check. The backend also exposes an atomic create-question-and-link command. Local undo, asynchronous saving, and durable commit are therefore connected operations with distinct boundaries.

8.3 Source preservation and raw fallback

The source adapter retains original wrappers, extension attributes, separators, and untouched regions where the supported contract permits safe reconstruction. Unsupported content remains editable raw source. A deliberately edited rich paragraph can be reserialized, while undo can restore its prior source spelling in the tested cases. The interface also supplies source and comparison modes.

This arrangement avoids making a visual editor the sole interpreter of every future extension. It also creates a responsibility to explain which regions are opaque and why a source-anchored action is unavailable. Ambiguous formatted selections are refused for operations that require an exact source anchor. The learner can use source view to choose the intended occurrence precisely.

DRAFT

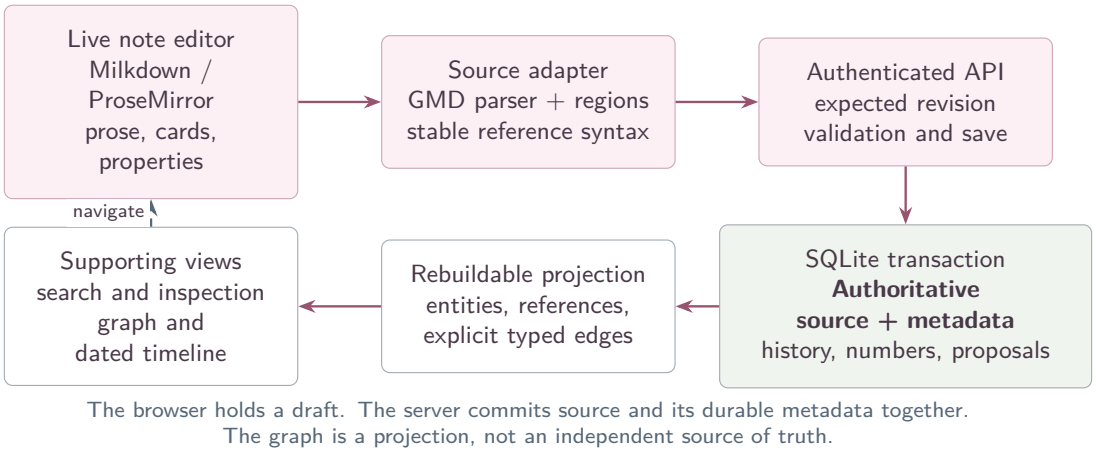

Figure 5: Hosted architecture in the supplied implementation. Shared GMD semantics connect browser authoring to the service, while SQLite retains source and durable metadata. Projection rebuilding does not replace backup of authoritative state.

8.4 Copy, move, and identity consequences

Copying a complete card into a context where its definition is already occupied needs fresh entity and property identities, with internal references remapped. Cutting and moving an absent definition can preserve identity. These cases have different semantic consequences even when the visible text is identical. The tested native clipboard behavior makes that distinction explicit within the hosted editor.

The behavior is local to the implemented environment. It does not define a distributed move protocol or guarantee every external editor's clipboard serialization. Future interoperability work should preserve the same distinction between duplication and continued identity, with diagnostics when the intended operation cannot be determined safely.

9 Revision Integrity and Durable Workspace State

Persistence is part of the semantic contract because a storage error can change what a note claims. A stale save can remove a qualification, a reused number can misidentify a question, and a projection rebuilt under the wrong source revision can display a relation that the author has not committed. The hosted architecture coordinates source, history, numbering, and projections through revision-checked database operations.

9.1 Durable and disposable state

The existing documents table remains authoritative for source, title, subject, author, revision, and timestamps. Additional durable tables retain source history, numbering allocations, monotonic counters, and proposals with review provenance. The projection table stores parsed entities, properties, occurrences, and diagnostics for a particular document and parser revision.

The projection is disposable; the entire database is not. A cache miss or invalid projection can be repaired from retained source without changing source or reallocating an existing number. Deleting the durable database would remove information that exact Markdown alone does not contain. Figure 5 locates these responsibilities in the hosted system.

DRAFT

9.2 Expected revisions and atomic updates

Writes, deletes, structured commands, and history restoration require an expected revision. A stale revision returns a conflict rather than overwriting the current document. No-op source and title updates create no additional revision. Restoring an earlier snapshot creates a new revision, preserving the intervening history.

Source changes, history snapshots, number allocation, and projection updates participate in the same SQLite transaction through a document observer. The supplied tests inject an indexing failure and check rollback of the coordinated changes. That result concerns this database boundary. It does not establish the multi-file journal recovery proposed for the earlier desktop system.

9.3 Asynchronous saves and recovery drafts

The browser serializes save requests and retains typing that occurs while an earlier request is in flight. A conflict preserves the competing source rather than silently selecting a winner. Recovery can create a copy, and the tested late-typing case retains a newer active draft while copying the earlier snapshot separately.

Browser-local storage is a best-effort recovery aid. It is distinct from a durable server revision and can be affected by browser storage conditions. The interface needs to communicate whether the visible text is saved, pending, conflicted, or retained locally. Otherwise, a graph rendered from an unsaved draft can be mistaken for a committed account.

9.4 Export and restoration

Exact GMD export reproduces stored document source. Readable Markdown export renders the material for ordinary reading. The full learning archive additionally carries subjects, documents, numbering and counters, history, and proposal provenance. These exports answer different recovery questions.

The implemented full-archive restore accepts a supported archive into an empty database, validates consistency, and rolls back a failed import. A subject-scoped snapshot is a reading export rather than a complete restoration package. Uploaded media, provider credentials, original legacy concepts, and other website domains remain outside the learning archive. Broader service recovery requires corresponding database and upload-directory backups.

9.5 Hosting and capacity limits

The prototype is an authenticated hosted system. Ordinary writing and local reflection prompts can work without a remote model provider, but that independence is not an offline desktop guarantee. The existing deployment's author labels do not constitute a generalized multi-tenant per-document access-control system.

New source, selections, titles, and archives have distinct code limits. These limits constrain input size; they are not measurements of usable capacity or latency. A larger-workspace study would need representative note lengths, graph sizes, network timing, and failure conditions. The current paper retains the implementation bounds in its technical appendix without presenting them as performance results.

10 Interpretive Proposals and the Review Boundary

The review mechanism separates suggested interpretation from committed writing. Its purpose is to preserve an accountable incorporation decision while keeping ordinary authoring independent of a remote provider. The mechanism is bounded: it supports selected operations against a source anchor, rather than a general autonomous agent that reorganizes the workspace.

DRAFT

10.1 Selected-source assistance

Proposal generation requires a nonempty selected passage and an expected document revision. A local path can generate a reflection prompt without contacting a model provider. A remote path requires explicit opt-in and transmits the selected text and optional instruction under a fixed request format. It does not send the entire workspace merely because a suggestion was requested.

The request boundary is testable with a mock provider. That test can establish the application payload in the exercised path. It does not determine a real provider's retention policy, confidentiality practices, factual accuracy, or educational value. Deployment needs an accurate account of the provider and the material a user is authorizing for transmission.

10.2 Proposal state and bounded operations

Suggestions are stored separately with their base revision, selected text, operation, and review status. The backend validates supported kinds and content limits. It rejects embedded GMD block markers or stable-reference syntax in provider prose, preventing a candidate's ordinary text from silently introducing executable structural commitments through that route.

A candidate question generated from within an existing valid entity can be represented as a create-question-and-link proposal. The candidate still contributes no source or graph effect until acceptance. The current response is bounded to at most three candidates. These limits constrain an implementation operation; they do not establish an optimal pedagogical number of suggestions.

10.3 Acceptance, rejection, and stale anchors

Acceptance checks the current revision, proposal base revision, and exact selected text. If the source changed while the provider was responding, the system does not silently re-anchor the suggestion to a later passage. A valid acceptance commits source, history, numbering, projection, and review status transactionally. Rejection records the decision without changing the source.

Figure 6 illustrates the distinction between pending material and committed effects. Acceptance records incorporation into the note. It does not certify the scholarly truth of the candidate, the quality of the question, or the depth of the learner's review.

10.4 Substantive judgment and review fatigue

A review button makes a decision explicit but can still become a routine click. Suggestions can be persuasive because of fluency, familiar vocabulary, or apparent confidence. A useful evaluation should therefore inspect reasons for acceptance and rejection, the detection of wrong targets or unsupported history, and whether the learner can explain an accepted change afterward.

The interface can support judgment by preserving the selected source, showing the proposed operation and its consequences, and permitting deferral. These affordances are design choices to evaluate. They do not eliminate the need for appropriate expertise or source consultation. A future system that supplies citations or cross-note synthesis would need additional source-verification and dependency contracts beyond the current selected-passage operation.

10.5 Provider-independent inquiry

The local authoring and deterministic projection paths remain useful without a remote model. This matters to the project's distinction between learning and generated content. A person can write an explanation, create a question, inspect its references, and revise the note through their own inquiry. Optional assistance is one component of the infrastructure, rather than the definition of generativity.

DRAFT

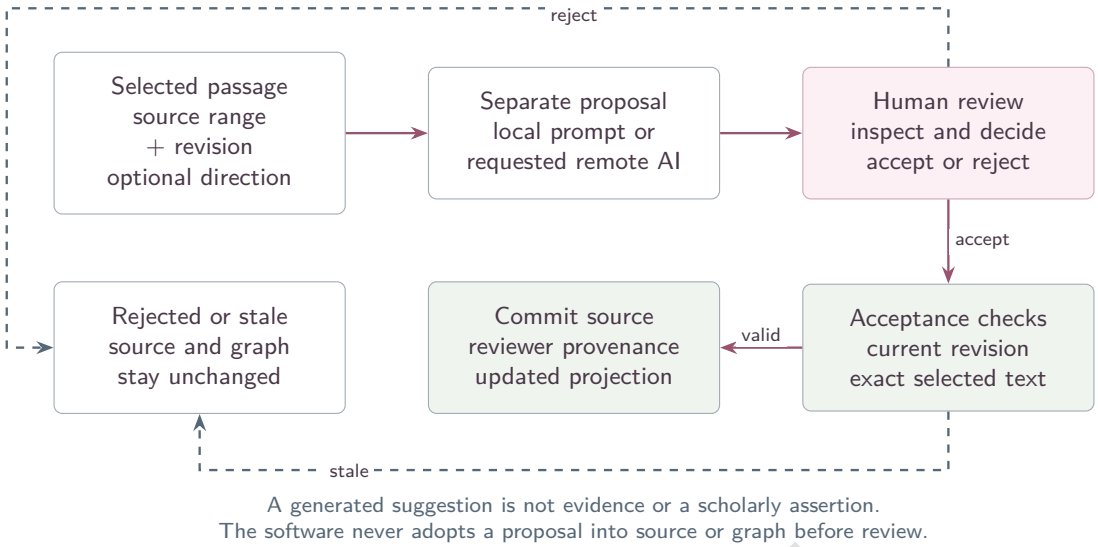

Figure 6: Proposal-review boundary in the hosted profile. Pending operations retain source anchors and remain separate from authored content. Acceptance requires a current, matching source context; rejection leaves source unchanged.

11 Graph, Historical Time, and Source Inspection

The supporting views provide alternative access to the same authored material. Their purpose is to help a learner inspect relations, locate source occurrences, and examine explicitly recorded time. Each view introduces selection and layout. The semantics of those transformations need to remain visible so that visual convenience does not become an unsupported substantive claim.

11.1 The graph as a source-derived view

The graph aggregates the eligible occurrences described in Section 7. Selecting an edge can expose its predicate and the note, entity, property, and range from which it arose. This route back to source is important when a displayed connection appears surprising. The learner can examine whether the predicate came from a typed property or an ordinary mention and can revise the actual source.

Figure 7 shows the synthetic five-entity graph and its provenance table. The layout is circular, and the line geometry carries no conceptual-distance or historical-order meaning. The table retains full titles where node labels are shortened. Multiple occurrences of one triple remain inspectable under the aggregated edge.

11.2 Explicit subject time

The hosted timeline reads valid, explicitly authored temporal-extent data in the supported display forms. It keeps undated entries separate. Document creation and modification timestamps do not supply historical subject time. A note written today can describe an earlier event, a long period, or an undated idea.

Figure 8 uses additional screenshot-only cards with arbitrary dates. The range 1900–1904 and day 1908-06-15 are display fixtures. They are not claims about the history of volume balance and are not part of the canonical five-entity specimen. The undated card demonstrates a retained absence of temporal information.

DRAFT

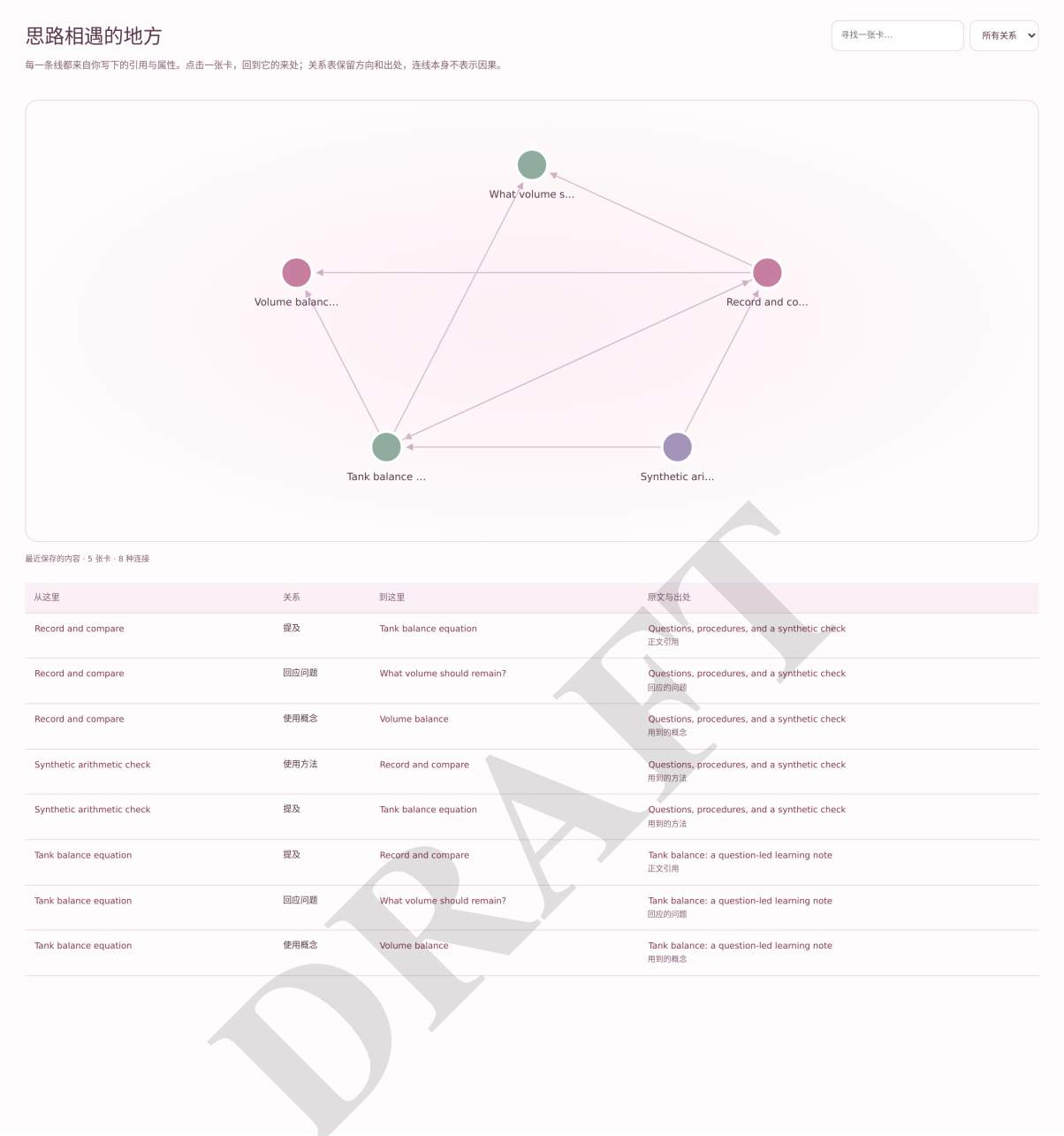

Figure 7: Actual source-derived graph for the five-entity water-tank specimen. Eight edges retain their distinct predicates and source occurrences. The provenance table provides a reading route independent of graph proximity.

11.3 Learning trajectories and interpretive limits

Revision history can help a learner revisit how an account changed. It can preserve an earlier question, rejected explanation, or revised transfer caution. The history records stored formulations and operations. It does not directly observe understanding, effort, confidence, or the circumstances of thought. A learner may understand a concept before recording it, or revise wording without changing their understanding.

The graph likewise offers a partial representation. Important relations can remain in ordinary prose or outside the notebook. An absent edge is not a missing cognitive connection, and a dense component is not a measure of mastery. A future learning-trajectory study should compare stored changes with task performance and participant accounts under appropriate consent, rather than infer mental development from interaction counts alone.

DRAFT

Figure 8: Explicit-time display with synthetic range, day-precision, and undated entries. Dates are arbitrary fixtures supplied for the screenshot. Recording timestamps are not used to invent historical chronology.

11.4 Relation to event-defined spacetime

The generalized GR application provides a different temporal construction: explicit event precedence, predecessor-closed cuts, and topology of applicable accounts at a selected boundary. The learning prototype does not implement that model. Its historical-time display is an explicit-date projection, not a general spacetime engine.

A future learning profile might represent episodes of reading, question formation, experiment, discussion, and reinterpretation with partial ordering. That extension would need to distinguish occurrence from recording, preserve uncertainty and privacy, and justify what a trajectory query means. It would also need to avoid turning a personal notebook into compulsory surveillance. The present design leaves that extension as a research interface.

12 Worked Specimen of Question-Centred Authoring

The worked specimen demonstrates how a small authored explanation becomes reusable relational structure. The example uses a stipulated volume balance for one container and interval. It contains no physical observations or historical claims. Its purpose is to expose source semantics, independent question identity, and the distinction between a typed reference and an annotation mention.

12.1 Objects and declared assumptions

The five entities are a question about remaining volume, a volume-balance concept, a formula, a recording-and-comparison method, and an arithmetic example. Thirteen independently identified properties supply role, assumptions, motivating questions, concepts used, and related context. Stable UUIDs identify the objects; the core-only specimen does not invent hosted display-number allocations.

DRAFT

Table 4: Complete edge inventory of the canonical synthetic specimen.

Source	Predicate	Target
Formula	addresses	Question
Formula	uses_concept	Concept
Formula	mentions	Method
Method	addresses	Question
Method	uses_concept	Concept
Method	mentions	Formula
Example	uses_method	Method
Example	mentions	Formula

The balance assumes a single container boundary, a single interval, a common volume unit, an incompressible-liquid approximation, recorded inflow and outflow, and exclusion of other volume-changing effects. Under these stipulated assumptions, Equation 2 is the bookkeeping relation examined in the note.

$$V_{\text{end}} = V_{\text{start}} + V_{\text{in}} - V_{\text{out}}. \tag{2}$$

Equation 2 has a clear role and boundary. If omitted processes such as evaporation matter, the note's model needs revision. The software stores that caveat as authored content; it does not infer the omitted physics or certify the model's adequacy.

The invented values are 10 L initially, 4 L inflow, and 1 L outflow, giving 13 L. A stipulated final value of 13 L produces a zero arithmetic residual. The calculation illustrates the notation and method. It is not experimental validation or evidence of learning.

12.2 Authoring sequence

The learner first writes the formula and an explanation of its role. A motivating-question property references the independent question. A used-concepts property references volume balance. The method describes recording quantities over the same boundary and interval, and it also addresses the question and uses the concept. The example then refers to the method and mentions the formula as an annotation.

The question can be reused by a later method with a different observation strategy. Its identity remains independent of the formula that first referenced it. If the learner decides the question was too broad, a revised title can clarify its scope while stable references persist. If the inquiry is substantively different, a new question identity can preserve the distinction rather than silently merging it by title similarity.

12.3 Derived edge inventory

Table 4 gives the eight aggregated edges produced by the actual module. Figure 9 provides a vector representation from the handoff. Both are projections of the same explicit source semantics.

The final two edges arise from one valid reference-list bullet. Its first reference points to the method and receives uses-method. A later reference to the formula occurs in the annotation and remains mentions. Assigning uses-method to both would wrongly classify the formula target and overstate the author's declared relation. The specimen was chosen to make this boundary observable.

12.4 Revision and transfer cautions

A learner considering a transfer to population, energy, or budget accounting would need to examine what the quantities represent, how the boundary is chosen, and whether the same conservation or bookkeeping assumptions apply. The shared algebraic shape alone is insufficient. The note can preserve a proposed mapping and a transfer caution without turning the proposal into an established equivalence.

An incorrect typed target, a duplicate definition, or a malformed reference property can also be introduced deliberately as a technical countercase. The expected outcome is a diagnostic and withheld

DRAFT

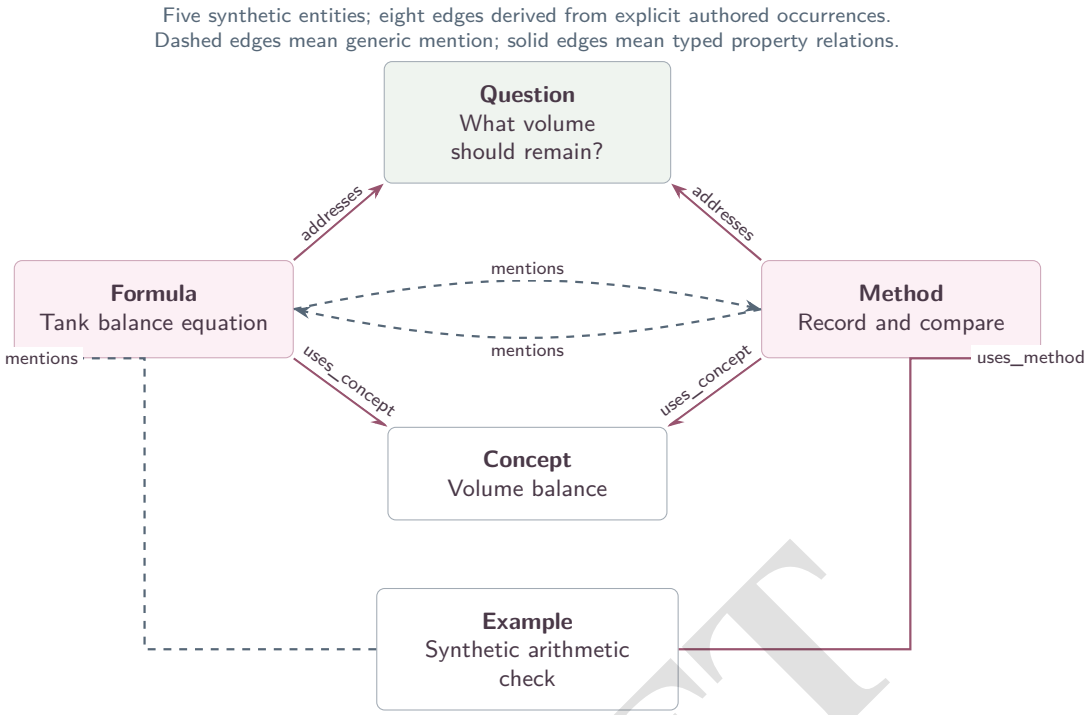

Figure 9: Vector view of the canonical specimen's authored relations. The graph distinguishes typed references from mentions; it does not encode physical causality or a learner's mental state.

semantic edge while source remains inspectable. These cases test the representation contract. A separate learner task would ask the participant to explain the diagnostic and decide whether to revise the source, choose another target, or leave the relation unresolved.

13 Engineering Evidence and Its Limits

The engineering evidence concerns whether the artifact preserves and applies its declared authoring commitments in tested cases. It is organized by representation, editor integrity, durable operations, and interface behavior. The supplied run and the paper's new subset execution are reported separately because they use different environments and overlap in test coverage.

13.1 Supplied hosted-system evidence

The handoff records 143 passing Node TAP tests, with no failures, cancellations, skips, or pending cases, at the specified hosted source commit. The total includes three parent browser-test aggregations. It is not a participant count or a set of 143 independent educational outcomes. The run used temporary databases, isolated hosts, and disabled or mocked providers.

The recorded environment used Linux and Node 18.19.1, with dependency versions retained in the handoff. Dependencies were supplied through an existing installation using `NODE_PATH`; a clean installation on an independent machine was not tested. The curated snapshot omits the full production server and private runtime state. These conditions qualify what reproduction of the supplied result requires.

13.2 Independent subset execution for this paper

The dependency-free GMD parser and live-source adapter suites were re-executed under the installed Windows Node environment. All 75 tests passed. The command, version, platform, exit code, and

DRAFT

raw TAP output are retained in the paper evidence directory. The cases cover parsing, literal contexts, unknown structures, reference semantics, identity ambiguity, Unicode boundaries, and source-preserving edits within the tested scope.

The new result does not rerun the hosted SQLite, HTTP, or browser suites. Its 75 cases overlap with the supplied 143 and are not added to them as independent coverage. Re-execution strengthens confidence in these particular source contracts across the recorded environments while leaving the broader deployment claims unchanged.

13.3 Failure-oriented coverage

The tests examine cases that can silently change meaning: a reference inside code becoming a real edge, duplicate identities resolving arbitrarily, a wrong-kind target receiving a typed predicate, a stale save replacing a newer source, or an indexing failure leaving numbering partly committed. Source-adapter cases also examine line endings, unknown attributes, opaque trailing content, and edits that would otherwise discard retained material.

The browser evidence includes direct writing, selection, undo, native clipboard behavior, composition events, conflict recovery, and supporting views. Simulated composition does not replace real operating-system input-method testing. A passing mobile viewport case establishes behavior in that scripted viewport, rather than comprehensive device coverage.

13.4 Security and preservation scope

The implementation displays raw HTML inertly, restricts actionable unsafe links, and limits live-canvas image loading to the same origin. These protections and their browser regressions concern specific paths. They do not constitute an independent security audit or establish suitability for every sensitive deployment.

Preservation claims are similarly scoped. No-op view switching and untouched regions retain exact source in the tested cases. Edited rich prose can acquire a different Markdown spelling. An archive can preserve learning metadata while excluding uploaded media. A graph can be rebuilt while the durable history and numbering tables remain necessary. The paper states these boundaries because a general claim of losslessness would obscure the actual contracts.

13.5 Evidential consequence

The results support the claim that the supplied prototype implements an inspectable, bounded authoring and persistence design. They do not establish learning gains, transfer effectiveness, user preference, latency at scale, all-platform accessibility, or live-provider quality. The appropriate next step is a staged evaluation whose measures correspond to those questions, developed in Section 14.

14 Prospective Evaluation of Authoring and Learning

The evaluation programme separates three questions: whether the software preserves authored commitments, whether people can use and interpret its affordances, and whether the resulting activity supports valuable learning. Each question has a different outcome and failure model. The programme below is a prospective design for research; no participant observations or effect estimates are reported.

14.1 Technical readiness and task validity

Before collecting learning outcomes, the study should establish a stable artifact version and representative authoring tasks. Technical readiness includes reproducible installation, isolated test storage, source and identity preservation, conflict handling, restoration, and reliable operation on the supported devices. The

DRAFT

existing tests provide a starting point, while real operating-system input methods, assistive technology, and adverse network conditions require further examination.

A task should require the intellectual distinction the feature is intended to support. Asking a participant merely to create five cards would measure compliance with a UI instruction. Asking the participant to explain which assumption limits a method and to identify where that assumption is documented can examine semantic understanding and source use. The two tasks have different evidential reach.

14.2 Formative authoring study

A first human study can observe a small, deliberately varied set of writing practices. Recruitment should describe participants' prior subject knowledge, tool experience, language, accessibility needs, and relationship to the design team. The study can include ordinary prose, mathematics, ambiguous target titles, unresolved material, and optional structure. Declining to create a card can be a reasonable response rather than an automatic failure.

The central task sequence begins with writing an explanation, promoting a selected passage, adding a role and motivating question, reusing or creating a question, continuing the original text, renaming the question, inspecting its references, and recovering from a save conflict. Measures include successful outcomes, loss of place, unintended mode changes, wrong bindings, recovery effort, and participants' explanations of what the software recorded.

Think-aloud sessions can reveal misunderstanding but may change the writing process. Their results should therefore be treated as formative evidence, with uninterrupted tasks examined separately if natural writing flow is an outcome. A fast click sequence accompanied by a false interpretation is an important design failure even when the server state is valid.

14.3 Semantic comprehension study

A focused comprehension task can present a mention, an addresses relation, a citation, and a historical-context link. Participants explain what each permits them to conclude and inspect the source occurrence. Deliberately misleading examples can test whether a person treats cites as support, addresses as complete solution, or graph proximity as conceptual similarity.

Scoring should use explicit rubrics with acceptable uncertainty and qualified answers. The study should report disagreement among assessors. A participant who correctly says that a relationship is not established can demonstrate better judgment than one who confidently infers a causal history. Appendix C supplies draft tasks and rubric anchors for piloting.

14.4 Comparative learning study

After formative problems are addressed, a comparative study can examine ordinary Markdown, manually structured notes, and the integrated relational editor. The conditions should provide comparable learning materials, source access, time, and training. Prior knowledge, tool familiarity, language, and task difficulty need explicit treatment. Repeated tasks require counterbalancing or matched materials to reduce order and practice effects.

The study's estimand should identify the activity, population, time horizon, and comparison. For example, it can concern the quality of a later explanation of a studied method's role and assumptions. That is narrower and more interpretable than a general claim that the application improves learning. Sample size and analysis should follow a pilot, a stated precision or power objective, and the available measurement properties; this paper supplies no invented effect size or sample requirement.

Outcomes can include explanation quality, warranted assumptions, useful new questions, source recovery, identified limitations, and qualified transfer. They should be assessed independently of the number of entities or links created. Assessors can be blinded to condition where the submitted answer permits it. The study should retain unstructured answers and negative results rather than rewarding resemblance to the application's preferred format.

DRAFT

14.5 Assisted performance and retained understanding

Performance with the notebook available is a legitimate outcome: an external record can help a person reason and act. Unaided understanding is a different outcome. The study should examine them separately. A participant may retrieve a correct answer quickly from a structured note while remaining unable to explain it without the tool. Another may create fewer links while developing a stronger internal account.

A delayed task can examine persistence, and a new problem can examine transfer. The delay and problem distance require justification before collection. Transfer should include cases where the appropriate response is to reject the mapping or request missing information. Otherwise, the outcome measure could reward overgeneralization rather than judgment.

14.6 Controlled evaluation of proposal review

The first review study can use controlled candidates with documented strengths and errors. This separates interface effects from changing model behavior. Cases can include a wrong target, a missing assumption, an unsupported historical claim, a superficial analogy, and a useful question that still requires revision. Participants should know that candidates may be wrong.

Measures include error detection, reasoned rejection, correction, appropriate deferral, and ability to explain accepted content. Acceptance rate alone is ambiguous: it can reflect useful candidates, weak scrutiny, or pressure to complete the task. A later live-provider study would record model and provider identifiers, prompts, transmitted scope, dates, outputs, validation failures, and independent assessments. Mocked provider tests are not that study.

14.7 Longitudinal use and sustainability

A longer study can examine whether structure remains useful after the novelty of the interface passes. Relevant observations include abandoned properties, unresolved references, maintenance effort, changing question formulations, export use, and the reasons a participant returns to plain prose or another tool. The study should also examine whether the notebook supports a continuing inquiry across disciplinary and language boundaries.

Personal notes can reveal uncertainties, private commitments, and sensitive relationships. Longitudinal collection therefore requires a proportionate data plan, participant control, and the ability to exclude material. Research access should not become a condition for using the notebook. Analysis can rely on synthetic tasks or participant-selected excerpts where those methods answer the question with less exposure.

14.8 Interpretation of outcomes

Possible findings include improved source recovery without improved unaided transfer, stronger explanations with greater authoring burden, or appealing graphs that increase false inference. Each finding would support a different revision. The design hypothesis is useful only if these results can change the implementation or its scope. The intended research outcome is a clearer account of which affordances help which activities under which conditions.

15 Ethics, Access, and the Governance of Learning Records

Learning records have value partly because they preserve unfinished thought. The same property can make them sensitive. A mistaken explanation, abandoned question, or private reflection should not automatically become a public performance indicator. The governance task is to preserve the learner's ability to use, revise, and leave the infrastructure under intelligible conditions.

DRAFT

15.1 Private inquiry and institutional assessment

A personal notebook and an institutionally assessed learning portfolio have different purposes. The current prototype supports hosted private authoring; it does not define a general educational assessment regime. An institution that later uses the records for grading, selection, or monitoring would introduce new power relations and evidence obligations.

The design should preserve room for tentative and incorrect writing. If every revision is interpreted as a failure or every unstructured paragraph as incomplete work, the infrastructure can discourage the exploration it seeks to support. Any consequential assessment profile would need explicit criteria, participant notice, a challenge route, and a distinction between recorded behavior and inferred understanding.

15.2 AI transmission and informed control

Selected-source transmission is a concrete boundary that a user can inspect. It reduces the amount sent relative to an entire workspace request, but it does not make sensitive text harmless. A deployment should communicate the configured provider and material scope accurately and allow ordinary use without remote assistance.

Review provenance also needs proportionate treatment. Retaining that a candidate was accepted can help reconstruct an account, while retaining every rejected private thought indefinitely may create unnecessary exposure. The current implementation records review states; a retention policy for sensitive deployment requires further design and qualified contextual review.

15.3 Accessibility and language

The notebook's intended writing practices include multilingual text and mathematics. Scripted composition events and a single browser family cannot establish real-world input-method or assistive-technology support. Future evaluation should include keyboard-only work, focus behavior, screen-reader interpretation, error recovery, and native Chinese and Japanese input where those users are in scope.

WCAG 2.2 supplies a relevant reference for web accessibility assessment [12]. The artifact does not claim present conformance. Beyond technical access, translated property labels need semantic review: a phrase such as motivating question can invite a historical-causal interpretation if its interface explanation is unclear. A glossary and source inspection help, but comprehension still requires evaluation.

15.4 Portability and dependence

Portability concerns more than readable text. Stable identities, numbering, history, proposal provenance, and media dependencies can affect later reconstruction. The full learning archive preserves more than a Markdown export, while still excluding some service state. A user needs to understand those differences when backing up or leaving a host.

An open source representation can reduce dependence without eliminating hosting and maintenance costs. The infrastructure's sustainability depends on who can repair it, interpret extensions, preserve backups, and update the export path. A future local implementation could change the failure and access model, but the present hosted system should be evaluated under its actual dependencies.

15.5 Metrics and epistemic authority

Entity counts, link density, accepted proposals, and time spent in the editor are easy to collect. Their meaning for learning is uncertain. Turning them into performance scores can create incentives to produce visible structure rather than warranted explanation. The design therefore treats these quantities as descriptive telemetry, if collected with an appropriate basis, rather than measures of mastery.

The same restraint applies to labels such as theorem, insight, or source. They are authored categories whose validity depends on the underlying content and inquiry. The software can preserve who committed a label and where it appears. It cannot grant scholarly authority through the label alone.

DRAFT

16 Interfaces with Generativity Epistemic Infrastructure

The relation between the learning artifact and the wider standards programme concerns reusable distinctions and future mappings. The systems currently have different representations, persistence boundaries, and application purposes. This section develops a research interface without claiming a delivered common protocol.

16.1 Shared distinctions and domain-specific categories

Both designs distinguish retained source from derived views, stable identity from display labels, recording time from represented time, and incorporation from substantive truth. Both preserve revision and expose selected grounds of a relation. These similarities support comparative design analysis.

The domain categories differ. A GMD question, formula, or method is an authored entity kind. The generalized programme's Entity, State, Event, Process, Relation, and Property are operational recording roles. A GMD method definition could be recorded as a document entity, while an actual execution of that method could require Event or Process accounts. Mapping the word method directly to Process would collapse the distinction between a description and an occurrence.

16.2 A prospective mapping contract

A mapping proposal should first identify its purpose. An archival export of a learning document might preserve its exact source as material, identify a document revision as a retained account, and represent selected source-derived triples through qualified Relation records. Each mapped edge would retain the original source occurrence and predicate scope. A cites edge would remain a citation relation, without becoming accepted evidence.

The mapping should also preserve numbering as presentation metadata and UUID binding as source identity. Unknown property keys and opaque source regions need retention with explicit limitations. A partial export should identify excluded documents and unavailable targets rather than manufacture complete referential closure. Reverse mapping needs to state which distinctions can be recovered and which require sidecar material.

16.3 Persistence and temporal differences

The learning service commits source and metadata in SQLite transactions. The generalized service publishes complete project snapshots through Git with transaction receipts. Both can support expected-version checks, but their retry, history, export, and administrator threat models differ. A shared conceptual notion of revision is insufficient to make their serialized operations interchangeable.

Temporal mapping is more delicate. The learning timeline reads explicit date properties. The generalized spacetime profile derives an event partial order and cuts from precedence accounts. A date range does not by itself supply the latter's event boundaries or asserted precedence. A converter should retain the date information and mark the missing event semantics, rather than infer a complete chronology.

16.4 Learning records as epistemic packages

A research-oriented learning note can contain a claim, source, interpretation, uncertainty, and later revision. Those elements resemble the programme's epistemic package concerns. A more explicit profile could help a learner distinguish source content from their own inference and identify where an accepted proposal changed the account.

The profile should remain proportionate to learning. Requiring a complete institutional evidence dossier for each provisional thought would impose a different purpose on the notebook. A layered design can let simple notes remain simple while offering richer structures for consequential or collaborative use. The relevant test is whether the added structure helps recover the learner's grounds without overwhelming the activity.

DRAFT

16.5 Development priorities

The immediate priorities are reproducible hosted deployment, real-device authoring and accessibility studies, semantic-comprehension evaluation, and independent review of source-preservation limits. Cross-domain mapping can follow a small set of fixtures before a general integration is attempted. The fixtures should include a renamed question, an unresolved external reference, a revised proposal, an ambiguous identity, and a dated note whose subject time differs from editing time.

The result could be a learning-specific profile within a broader infrastructure family. That status would require its own specification and assessment process. The present paper contributes the analysis and artifact basis from which such work can proceed.

17 Concluding Account of the Learning Infrastructure

The proposed learning infrastructure makes selected relational commitments available within ordinary writing. Independent questions can be reused across formulas and methods; stable references preserve binding through renaming; occurrence-backed edges expose the source of a displayed relation; and reviewed proposals remain separate from committed content. Revision integrity and explicit export boundaries support the continued use of those records.

The hosted prototype implements these responsibilities through a continuous editor, a conservative semantic core, retained source regions, and coordinated database operations. Its design differs materially from the earlier desktop proposal, and the paper preserves that distinction. The synthetic specimen and engineering evidence demonstrate bounded behavior while leaving educational effectiveness to the proposed studies.

Generative relational learning remains a research orientation whose value depends on what people can understand, question, revise, and appropriately transfer. The notebook offers means for those activities. Its success should be assessed through warranted explanation, recoverable grounds, meaningful judgment, and sustainable authoring effort, including evidence that would motivate changing or narrowing the design.

DRAFT

References

[1] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13. ACM, 2019. doi: 10.1145/3290605.3300233. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2019/01/Guidelines-for-Human-AI-Interaction-camera-ready.pdf. Accessed 30 September 2026.

[2] Michelene T. H. Chi and Ruth Wylie. The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. *Educational Psychologist*, 49(4):219–243, 2014. doi: 10.1080/00461520.2014.965823. URL https://education.asu.edu/sites/default/files/lcl/chiwylie2014icap_2.pdf. Accessed 30 September 2026.

[3] Douglas C. Engelbart. Augmenting Human Intellect: A Conceptual Framework. Technical Report AFOSR-3223, Stanford Research Institute, 1962. URL https://dougengelbart.org/content/view/138/. Accessed 30 September 2026.

[4] Generativity Standards Program contributors. Programme Brief: Why, What, and How We Develop the Generativity Standards. Supplied project source, 2026. Version 0.5; preliminary programme brief.

[5] Dedre Gentner. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7 (2):155–170, 1983. doi: 10.1207/s15516709cog0702_3. URL https://groups.psych.northwestern.edu/gentner/papers/Gentner83.2b.pdf. Accessed 30 September 2026.

[6] Wanhong Huang. Generative Relational Learning Paper Handoff. Supplied project source, 2026. Private authoring handoff, 29 September 2026; hosted source f40c74c9e209218abc5701cc74a44a57b0db185f.

[7] Wanhong Huang. Epistemology of Generative Relational Economics. Supplied project source, undated. GRE Chapter 5; FM-005; supplied manuscript.

[8] Joseph D. Novak and Alberto J. Cañas. The Theory Underlying Concept Maps and How to Construct and Use Them. Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition, 2008. URL https://cmap.ihmc.us/docs/theory-of-concept-maps. Accessed 30 September 2026.

[9] ProseMirror Project. ProseMirror Guide. URL https://prosemirror.net/docs/guide/. Accessed 30 September 2026.

[10] SQLite Project. Atomic Commit in SQLite. URL https://www.sqlite.org/atomiccommit.html. Accessed 30 September 2026.

[11] Merlin C. Wittrock. Learning as a Generative Process. *Educational Psychologist*, 11(2):87–95, 1974. doi: 10.1080/00461527409529129. URL https://www.tandfonline.com/doi/abs/10.1080/00461527409529129. Accessed 30 September 2026.

[12] World Wide Web Consortium. Web Content Accessibility Guidelines (WCAG) 2.2. Recommendation, 12 december, W3C, 2024. URL https://www.w3.org/TR/2024/REC-WCAG22-20241212/. Accessed 30 September 2026.

DRAFT

A Hosted Protocol Reference and Registry Inventory

This appendix summarizes the implemented hosted profile and includes registry inventories generated from the supplied CSV artifacts. It is a research reference for the inspected version, rather than a newly adopted Standard. The canonical module and source hashes remain identified in the evidence package.

A.1 Syntax and layer-specific bounds

Entity identities use `e_` followed by a lowercase hexadecimal UUID-shaped string; properties use the corresponding `p_` prefix. The parser validates shape rather than restricting accepted source to a particular UUID version. New identities are generated as UUIDv4. Required entity attributes are identity, type, and title. Property attributes identify identity and key.

Canonical references bind to the stable identity and can include an escaped authored label. Number-shaped shorthand remains unbound. A workspace-qualified target is recognized syntactically but remains unavailable to the hosted resolver. Properties use Markdown, reference-list, or structured mode as declared by the registry.

The parser bounds an opening tag to 16 KiB and entity titles to 500 Unicode code points as counted by its JavaScript operation. Hosted commands impose a narrower 240 UTF-16-unit title bound. Reference-label source has both a 2 KiB UTF-8 bound and a 2,048-unit string bound. New document source is limited to 1 MiB, while selected proposal text is limited to 12,000 UTF-16 units. These are distinct validation layers, not one universal character limit.

The full learning archive has a 20 MiB restore bound and at most 50,000 records in each archive array. Restoration targets an empty database and validates the durable relationships before commitment. These bounds do not establish measured latency or capacity at their limits.

A.2 Entity registry

Table 5 retains the 28 registered entity kinds and their intended purpose. Registry availability is distinguished from the validity of an authored claim. A kind label supplies organization and interface behavior, while the underlying scholarly judgment remains attributable to the author.

Table 5: Hosted entity-kind registry, retained from the inspected source.

Kind	Purpose	Distinction
question	An articulated inquiry.	Its topic, answer, and motivating situation are distinct.
problem	A problematic situation or practical challenge.	A problem need not be a question sentence.
concept	A reusable conceptual distinction or object.	A concept may have several definitions.
definition	A particular formulation of meaning or criteria.	A definition does not exhaust the concept.
formula	A mathematical expression with interpretation.	Type assignment does not validate the mathematics.
theorem	A statement presented with proof-related support.	Application labels do not certify a proof.
proposition	A structured statement in the author's reasoning.	No universal strength ranking is imposed.
hypothesis	A conjectural explanatory or testable proposal.	Its category is separate from its support status.
argument	A case involving premises, inference, and conclusion.	A collection of citations is not automatically an argument.
insight	An interpretation or connection worth preserving.	An insight label is not a truth judgment.

DRAFT

Kind	Purpose	Distinction
method	A procedure for inquiry, analysis, or intervention.	Research and practical use can coexist.
history	A historical event, period, development, or context.	Subject time is distinct from edit time.
claim	A specific asserted statement.	A claim may be supported, contested, or quoted.
observation	A situated encounter or measurement account.	Observation is not context-free access to truth.
model	A representational construction.	A model can contain multiple formulas.
practice	An organized practical activity.	It can use several methods.
assumption	A reusable condition taken for a purpose.	It is not automatically an empirical claim.
limitation	A boundary on validity or transfer.	It need not invalidate the entire method.
example	An illustration of a conceptual structure.	Illustration does not establish generality.
case	A bounded empirical or practical instance.	A case can support several interpretations.
analogy	A proposed cross-context correspondence.	Analogy is not identity or equivalence.
source	Bibliographic or evidentiary material.	Availability and evidential reliability are distinct.
dataset	A defined collection of observations or records.	Dataset identity is separate from its file location.
discipline	An intellectual field or tradition.	It need not exclusively own its concepts.
debate	A contested inquiry and its positions.	Grouping does not choose a winner.
person	A participant relevant to the notes.	Private details require privacy care.
organization	An institution or collective actor.	Names and institutional continuity can differ.
artifact	A relevant created object or work.	A local attachment is only one representation.

A.3 Property registry

Table 6 retains the 31 property keys and their modes. The relation-bearing subset is explained in Table 3. Structured JSON syntax does not imply full validation of every domain-specific temporal or bibliographic convention envisaged by the earlier desktop proposal.

Table 6: Hosted property registry and semantic modes.

Key	Mode	Description
role	markdown	What intellectual or practical work does this object perform?
motivating_questions	references	Link or create independently defined questions.
related_concepts	references	Neutral association with a concept; does not assert use.
used_concepts	references	Concepts explicitly used by this object.
related_methods	references	Neutral association with a method; does not assert use.
used_methods	references	Methods explicitly used by this object.
related_questions	references	Other relevant questions; does not imply hierarchy.
assumptions	markdown	Narrative conditions and caveats; refs are mentions.
assumption_entities	references	Explicit reusable assumptions.
limitations	markdown	Boundaries of validity and use.

DRAFT

Key	Mode	Description
history	markdown	Historical prose; inline references alone are mentions.
historical_contexts	references	Explicit context assignment, not causation.
sources	references	Sources cited, with locators in a separate property.
source_locators	structured	Edition/page/section/equation locators; no invented values.
derivation	markdown	Reasoning leading to a formulation.
proof	markdown	Proof text or reference; not an automated certificate.
premises	markdown	Premises of an argument.
conclusion	markdown	Conclusion as authored.
objections	markdown	Challenges and counterarguments.
responses	markdown	Responses to objections.
transfer	markdown	A proposed reusable structure and target context.
transfer_cautions	markdown	Important non-equivalences and assumption failures.
temporal_extent	structured	Explicit time data; never inferred from edit timestamps.
aliases	structured	Search labels, represented as a JSON array of strings.
inputs	markdown	Required information or material.
outputs	markdown	Expected output and interpretation.
procedure	markdown	Steps of inquiry or intervention.
applications	markdown	Narrative use cases.
examples	markdown	Illustrations with optional entity references.
arguments	markdown	Positions and their reasoning.
bibliography	structured	Source metadata as stated; missing values remain missing.

A.4 Operation and diagnostic boundaries

The implemented operations include creating entities and properties, stable reference generation, title changes, source patches, explicit question creation and linking, source save and restoration, export, and bounded proposal review. Generic arbitrary relation editing and a complete standalone assertion language remain outside the hosted operation set.

Diagnostics retain unresolved and ambiguous material rather than choosing an arbitrary interpretation. Literal contexts, unsupported source structure, wrong target kinds, duplicate identities, and unavailable external workspaces have different causes. Their effects on graph eligibility and rich editing can differ. The source remains the route for examining those distinctions.

B Canonical Synthetic Specimen

The complete source below is the handoff's five-entity water-tank specimen. It is reproduced so that the edge inventory in Section 12 can be inspected directly. It contains thirteen properties and produces five valid nodes and eight aggregated edges with no diagnostics in the clean specimen. The arbitrary-date screenshot fixtures are separate and are not included here.

The notation inside the source is intentionally literal GMD and Markdown. Display mathematics inside the listing belongs to the retained specimen rather than to the paper's numbered formal constructions. The source identifiers are stable test identities and have no connection to private participants or actual physical objects.

```
# Synthetic water-tank bookkeeping
```

DRAFT

```
This editable GMD specimen demonstrates explicit authoring and graph derivation.
All entities and numbers below are illustrative. No historical date is supplied.

<:entity id="e_10000000-0000-4000-8000-000000000001" type="question" title="What volume should remain?">
How can we account for a tank's stored liquid over one fixed interval?
This is a synthetic teaching question, not a report of an experiment.
<:property id="p_20000000-0000-4000-8000-000000000001" key="role">
Organize an inquiry into boundaries, recorded flows, and an interpretable check.
</:property>
</:entity>

<:entity id="e_10000000-0000-4000-8000-000000000002" type="concept" title="Volume balance">
A volume balance compares storage at the boundaries of a specified interval with recorded inflow and
    outflow.
It is a bookkeeping model under stated assumptions.
<:property id="p_20000000-0000-4000-8000-000000000002" key="aliases">
["storage bookkeeping", "tank volume balance"]
</:property>

<:property id="p_20000000-0000-4000-8000-000000000003" key="transfer">
Try an analogous bookkeeping structure for an inventory, then state what counts as an addition, removal,
    and stored quantity.
</:property>

<:property id="p_20000000-0000-4000-8000-000000000004" key="transfer_cautions">
The analogy does not establish physical equivalence. Units, unrecorded flows, and volume-changing effects
    must be reconsidered.
</:property>
</:entity>

<:entity id="e_10000000-0000-4000-8000-000000000003" type="formula" title="Tank balance equation">
For a fixed interval in the stated model:

$$
V_{\mathrm{end}} = V_{\mathrm{start}} + V_{\mathrm{in}} - V_{\mathrm{out}}
$$

A worked procedure is described by [::@e_10000000-0000-4000-8000-000000000004]. This sentence is a mention.
<:property id="p_20000000-0000-4000-8000-000000000005" key="role">
Make the bookkeeping relationship explicit and identify the quantities that need to be recorded.
</:property>

<:property id="p_20000000-0000-4000-8000-000000000006" key="motivating_questions">
- [::@e_10000000-0000-4000-8000-000000000001]
</:property>

<:property id="p_20000000-0000-4000-8000-000000000007" key="used_concepts">
- [::@e_10000000-0000-4000-8000-000000000002]
</:property>

<:property id="p_20000000-0000-4000-8000-000000000008" key="assumptions">
Use one container boundary, one time interval, and a common volume unit. Treat the liquid as incompressible
     for this example. Record all inflow and outflow; exclude other volume-changing effects from the model
    .
</:property>
</:entity>

<:entity id="e_10000000-0000-4000-8000-000000000004" type="method" title="Record and compare">
Use [::@e_10000000-0000-4000-8000-000000000003] as an explicitly stated bookkeeping rule; the reference in
    this paragraph is only a mention.
<:property id="p_20000000-0000-4000-8000-000000000009" key="motivating_questions">
- [::@e_10000000-0000-4000-8000-000000000001]
</:property>
```

DRAFT

```
<:property id="p_20000000-0000-4000-8000-000000000010" key="used_concepts">
- [::@e_10000000-0000-4000-8000-000000000002]
</:property>

<:property id="p_20000000-0000-4000-8000-000000000011" key="procedure">
1. Specify the container boundary and interval.
2. Record starting volume, total inflow, and total outflow in the same units.
3. Compute the model's expected final volume.
4. Compare with a stated final value and discuss any discrepancy without inferring its cause automatically.
</:property>
</:entity>

<:entity id="e_10000000-0000-4000-8000-000000000005" type="example" title="Synthetic arithmetic check">
Use invented values of 10 L initially, 4 L inflow, and 1 L outflow. The equation gives 13 L. A stipulated
     final value of 13 L gives a zero arithmetic residual.
This example tests interpretation of the notation; it is not a measurement or a learning evaluation.
<:property id="p_20000000-0000-4000-8000-000000000012" key="used_methods">
- [::@e_10000000-0000-4000-8000-000000000004] -- compare the rule [::@e_10000000
     -0000-4000-8000-000000000003] as an annotation.
</:property>

<:property id="p_20000000-0000-4000-8000-000000000013" key="limitations">
A zero residual in invented numbers establishes no empirical accuracy, completeness of records, or
     educational benefit.
</:property>
</:entity>
```

The full graph JSON preserves occurrences and ranges, and the compact diagram data supports the vector figure. A reproduction should compare the original source and parser hashes as well as node and edge counts. Matching counts alone could conceal a wrong predicate or rebound target.

C Draft Tasks and Rubric Anchors for Future Studies

These materials operationalize the evaluation programme in Section 14. They are proposed instruments requiring pilot refinement, participant review where appropriate, and an explicit analysis plan. No reliability coefficient, learning effect, or participant result is claimed.

C.1 Authoring task

The participant receives a short explanation of a method, its assumptions, and two similarly named questions. The task is to write a note in ordinary prose, make one selected passage independently addressable, and connect it to the appropriate question. The participant may decide that neither existing question fits and create a distinct one. The observer records preservation of the original explanation, target binding, interruption, recovery, and the participant's account of the resulting relation.

A subsequent task asks the participant to rename the question and find its references. Success requires recognizing that the title changed while the stable target remained. A separate duplication task distinguishes a copied card from a moved card. The observer checks both software state and the participant's intended identity operation.

C.2 Semantic interpretation task

The participant is shown four relations: a prose mention, a motivating-question link, a citation, and a historical-context link. For each, the participant identifies what the source explicitly commits to and what would require additional evidence. A source-inspection action then asks the participant to locate the occurrence that generated the edge.

DRAFT

Table 7: Draft rubric dimensions for explanation and qualified transfer.

Dimension	Strong anchor	Weak or adverse anchor
Role	Connects the method to a specific inquiry and purpose	Repeats notation without explaining its use
Assumptions	Identifies a premise and the consequence of its failure	Lists irrelevant conditions or treats the rule as universal
Grounds	Locates a relevant source or declared stipulation	Treats a valid graph edge as evidence of truth
Transfer	States a mapping and a material limit	Relies only on similar symbols or vocabulary
Question formation	Identifies a further uncertainty grounded in the case	Adds an unrelated generic question
Revision	Explains a change in relation to earlier grounds	Changes wording without addressing the identified error

Rubric anchors distinguish an accurate scoped explanation, a partially accurate explanation with a material omission, an unsupported stronger inference, and inability to interpret the relation. A response that correctly retains uncertainty receives credit. For example, identifying that a citation does not establish support is a successful interpretation rather than an incomplete answer.

C.3 Explanation and transfer task

After studying the synthetic balance note, the participant explains the role of the relation, identifies one assumption, and describes a circumstance requiring revision. A transfer task supplies a superficially similar expression in another domain and asks which correspondences hold, which conditions differ, and what evidence would justify use.

Table 7 gives draft scoring dimensions. The dimensions should initially be reported separately. Combining them into one score would require a justification of weighting and interpretation; a high total could otherwise conceal a consequential error.

C.4 Proposal-review task

Controlled proposals include a useful clarification, a wrong target, a missing assumption, and an unsupported historical claim. The participant can accept, revise, reject, or defer. The study records reasons and later explanation of incorporated content. A candidate's fluency and usefulness should be varied separately where feasible, so that the study can examine unwarranted confidence rather than merely compare obviously good and obviously bad prose.

C.5 Reporting and adverse outcomes

The study report should identify the artifact version, recruitment, task materials, training, assistance, exclusions, missing data, assessor disagreement, and the scope of any inference. Adverse outcomes include lost text, wrong bindings, misunderstood predicates, review fatigue, avoidance caused by burden, and endorsement of unsupported transfer. The design can be revised in response to these findings without treating non-use of optional structure as a participant deficiency.

D Artifact Inventory and Reproduction

The supplied archive hash is recorded in `papers/INPUT_PROVENANCE.json` in the programme source tree. The learning handoff identifies the hosted source commit, curated implementation files, dependency lock,

DRAFT

synthetic specimen, figures, screenshot fixtures, and successful execution output. Its original manifest supports byte-level comparison of the package.

The source and graph specimen are retained in this paper's `sources/protocol/` directory. Registry CSVs and the source/test evidence index are also retained. The four screenshots are authentic captures of isolated synthetic notes. Capture metadata records dimensions, fixture content, and hashes. Original vector figures are supplied as compiled PDFs, with editable TikZ sources in the handoff.

The paper's independent subset run used Windows Node v22.17.0 and the command below from the supplied implementation directory. It reported 75 passing cases and no failures.

```
node --test test/curiosity-gmd.test.js test/curiosity-live-source.test.js
```

The raw result is `evidence/rerun-core-tests.tap`; environment and scope are in `evidence/rerun-core-tests.json`. The handoff's broader 143-test output is preserved separately. A fresh full deployment or clean dependency installation is not claimed by the subset run.

The manuscript can be compiled with XeLaTeX and BibTeX through latexmk. The source bundle includes the required figures, bibliography, section files, and generated registry tables. The build and structural check reports identify unresolved-reference checks, figure/table/equation cross-references, page counts, and inspected layout samples. These checks support a reproducible reading artifact while leaving educational evaluation and comprehensive accessibility assessment to further work.

The handoff remains a private authoring source. The paper uses its conceptual and technical material under the user's instruction to prepare the manuscript. It does not publish the original design dialogue to an external service or convert the curated source snapshot into a claimed public dataset. Any later release should preserve the included notices and the distinction between project-authored material and third-party components.

DRAFT