

groundworkers: Concept Grounding and Graph Traversal for Agents

Georgina Kennedy^{1,2}

1 School of Clinical Medicine, University of New South Wales, Sydney, Australia,

2 Ingham Institute for Applied Medical Research, Sydney, New South Wales, Australia,

Background

OMOP vocabulary handling applications need the same operations in many different contexts - concept lookup, standard mapping, hierarchy traversal, relationship inspection, and some form of fallback when lexical matching fails (e.g. normalisation, string edit distance measures or embedding search). This functionality supports vocabulary curation tools, source data mapping, cohort design support, and increasingly in LLM-assisted workflows. None of these operations are especially novel, however they are often re-implemented separately inside each application, with slightly different assumptions about how to search, when to broaden, how to treat non-standard concepts, and what counts as an acceptable mapping candidate.

As vocabulary operations move closer to full or partial automation, or agent-assisted workflows, it becomes more important to have a coherent architecture underpinning this functionality. Variability is not inherently a problem, as in imperfect clinical terminology settings different retrieval and resolution strategies can be complementary, and a system that supports multiple paths may perform better than one that enforces a single prescriptive route. The more important requirement is that these paths are explicit. When a human reviewer and an agent-assisted pipeline disagree, or multi-agent mixture of expert-style pipelines produces contradictory results, it should be possible to tell whether that disagreement arises from genuine semantic ambiguity, from different evidence channels, or from a true implementation error. Without a shared and clearly signalled strategy boundary, those cases are difficult to separate.

Source-to-standard mapping is a common setting in which this problem becomes operationally visible. Preparing a data dictionary, registry coding scheme, or local schema for OMOP analysis usually involves resolving institution-specific labels, local codes, abbreviated clinical terms, and semi-structured descriptions to standard concepts. This work is repetitive and depends on a combination of lexical search, cross-vocabulary mapping, hierarchy-aware backoff, and selective human judgment.

Methods

This work describes a four-layer open-source stack that separates deterministic vocabulary behaviour from workflow orchestration.

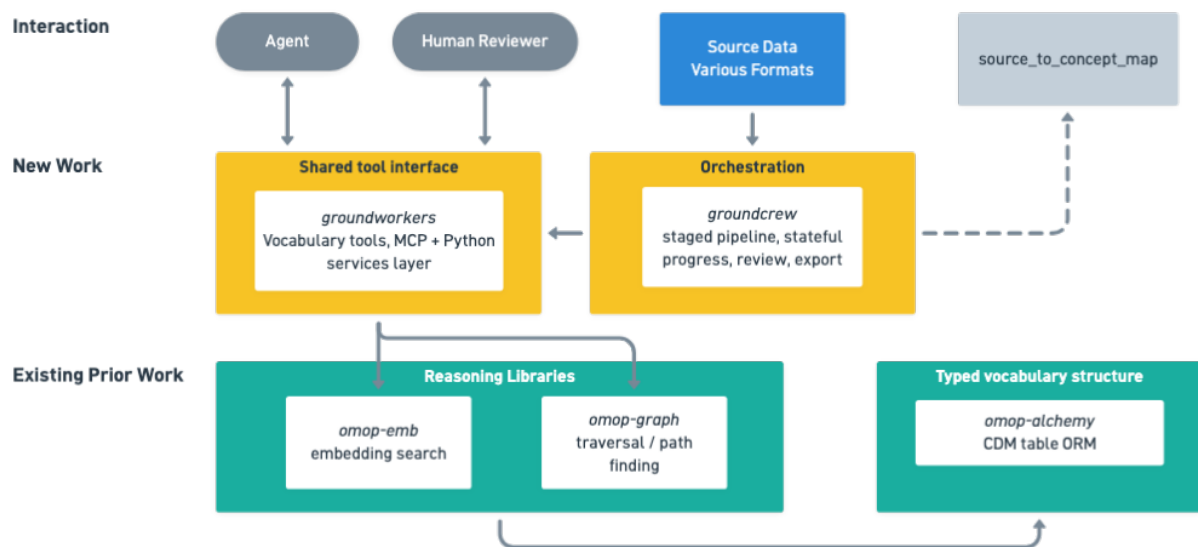
Layer	Component	Why it is important
Typed vocabulary structure	omop-alchemy	CDM vocabulary tables as a strongly specified ORM; types and relationships legible to both software and agents
Reusable reasoning libraries	omop-graph; omop-emb	Graph traversal, path analysis, and semantic retrieval built on the typed model
Shared tool interface	groundworkers	Python library and MCP server exposing lookup, mapping, traversal, grounding, and evidence assembly
Workflow orchestration	groundcrew	Acts as the ACP layer for ingestion, staged grounding, session state, review, escalation, and export.

`groundworkers` packages graph and embedding capabilities into a single callable vocabulary service. The same operations are available directly from Python via a services layer or remotely

through MCP and HTTP. Concept lookup, traversal, semantic retrieval, and mapping support are assembled once and reused across scripts, interactive tools, and automated pipelines.

`groundcrew` sits above this and uses those operations to run end-to-end source-to-standard mapping workflows. It accepts source material in diverse formats (CSV data dictionaries, SQL DDL, JSON schema, registry codings, and free text) and works each item through a seven-stage grounding pipeline: exact label lookup, full-text search, embedding similarity, cross-vocabulary mapping, hierarchy expansion, and optional LLM inference. Sessions are fully durable. Thresholds, escalation rules, and review policy live in this layer, where they can be adjusted without touching the underlying vocabulary operations.

The stack is packaged for local deployment through `omop-groundwork`, which bundles the required services into a single workspace without external infrastructure dependencies.



Results

The main result remains architectural at this time, offering innovation through the distribution of concept grounding behaviour across four explicit layers rather than collapsed into a single application. The typed vocabulary model signals what exists; the graph layer offers reasoning paths and semantic retrieval; the MCP layer offers direct agentive tools as well as a services wrapper, and the ACP layer signals how those calls are coordinated into a task. This separation does not require every consumer to use the same resolution path. Instead, it allows multiple strategies to coexist over a common set of tools and assertions. One use-case may be a better fit for exact-first resolution, another may lean more heavily on embeddings or graph expansion, and a review workflow may explicitly compare several channels side by side. The shared lower layers provide common assertions, explicit evidence, and a clear place to inspect why strategies converge or diverge.

In applied source-to-standard workflows, the staged pipeline short-circuits on the earliest sufficient channel. Straightforward clinical labels can resolve through exact or full-text methods; abbreviations, informal terms, and other lexical edge cases can fall through to embedding retrieval; and items with no acceptable direct match can be broadened or contextualised using graph and mapping tools before escalating further. Optional LLM inference is therefore positioned as a residual reasoning step rather than as the primary mechanism of vocabulary grounding.

The same layered design also supports review and escalation processes. Automated processing, human validation, and autonomous review agents all operate over the same `groundworkers` tool interface, rather than over separate interfaces with unclear or subtly different semantics. This means acceptance, rejection, or revision of a candidate mapping happens against the same evidence bundles and traversal outputs used upstream, while still allowing different decision policies to sit above them.

Low-confidence outputs are not treated as final simply because a candidate exists; they are surfaced for review through the same shared tool layer used by the automated stages.

This structure is especially useful for source data onboarding because it produces outputs that are easier to reproduce and interrogate. Mapping candidates can be traced to lexical evidence by graph traversal path traces, identity-based mappings, hierarchy relationships, or embedding neighbours exposed through explicit tools, instead of being recoverable via specific application logic.

Conclusion

The primary contribution here is architectural, as `groundworkers` is infrastructure, not a system. It does not perform grounding decisions on its own, manage workflow state, or make claims about clinical appropriateness. Its contribution is to provide a stable, consistently-behaved vocabulary tool API that multiple tools and agents can share without re-implementing the underlying resolution logic for each new application. `groundcrew` wraps those operations into a durable workflow for end-to-end source-to-standard mapping, but the same tool surface equally supports lighter consumers like interactive vocabulary tools, standalone scripts, or agent-facing services that need vocabulary operations without session management or the full ingestion pipeline.

The stack is locally deployable, operating directly over existing CDM vocabulary tables. As OMOP-based tooling increasingly incorporates LLM and agentic components, consistent vocabulary behaviour across tools becomes harder to maintain without a shared foundation, which is the gap this stack is intended to fill.

References

1. Australian Cancer Data Network. `groundworkers` [Internet]. GitHub repository. Available from: <https://github.com/AustralianCancerDataNetwork/groundworkers>
2. Australian Cancer Data Network. `omop-graph` [Internet]. GitHub repository. Available from: <https://github.com/AustralianCancerDataNetwork/omop-graph>
3. Australian Cancer Data Network. `omop-emb` [Internet]. GitHub repository. Available from: <https://github.com/AustralianCancerDataNetwork/omop-emb>
4. Australian Cancer Data Network. `omop-groundwork` [Internet]. GitHub repository. Available from: <https://github.com/AustralianCancerDataNetwork/omop-groundwork>