

GRAPH RESEARCH LABS

AI-Assisted Ontology Engineering

Guardrails and Workflows for Using LLMs Safely



Graph
Research
Labs

July 2026

Dougal Watt · Graph Research Labs

Portable. Works with any LLM.

01

Claude

Your own account

02

Prompt set

In the GitHub repo

03

Audit tools

GRL ontoLinter, reasoner, SHACL

This is not a model-quality problem - it's a guidance problem. Typical mistakes LLM's make are below:

01 Complex class hierarchies

Deep, baroque trees that subclass everything in sight.

02 Inconsistent taxonomies

The same concept in three places under three names.

03 Uncontrolled property growth

Parallel inventions where the upper ontology already provides.

04 Modelling anti-patterns

Plan/occurrence collapse, role/type confusion, magnitude misuse.

LLM's have been trained on public ontologies of variable quality - this means they will make mistakes.

Maturity model: three levels of LLM extension from skills, to plugin, to harness

Skill

a focused prompt

The smallest unit of LLM extension. A reusable instruction set.

WHEN

Occasional manual use.

EXAMPLE

Each of the 7 prompts in your handout is a skill.

Plugin

an orchestration of skills

Workflow that fixes a problem, regenerates downstream artefacts, re-runs validation in one continuous flow.

WHEN

Repeated workflow.

EXAMPLE

GRL ontology modeller plugin.

Harness

a governed agent

Constrains an LLM agent to operate within an ontology and SHACL declarations. The full model-set is the program.

WHEN

Governance-bound autonomous operation.

EXAMPLE

GRL Semantic Agent Harness.

Maturity model level 1: skills

Skill

a focused prompt

The smallest unit of LLM extension. A reusable instruction set.

WHEN

Occasional manual use.

EXAMPLE

Each of the 7 prompts in your handout is a skill.

What this gives you: agent can describe

- *documented patterns*
- *domain expertise*
- *methodology*

You are an expert ontology engineer experienced in gist, BFO, and domain-specific upper ontologies. Reuse upper-ontology classes by IRI; do not redefine them. Your task is to generate competency questions for the ontology I am about to describe.].

Competency questions must be:

- Measurable: each question must have a clear pass/fail answer condition.
- Scoped: each question must be answerable using only the ontology and instance data.
- Useful: each question must reflect a real use case – something a downstream application or a domain user would actually ask.
- Distinct: no two questions should ask substantially the same thing.

For each competency question, provide:

1. The question in plain language.
2. The use case it serves.
3. The ontology elements (classes, properties) that would need to exist for it to be answerable.
4. A draft SPARQL skeleton (do not worry about correctness; sketch the shape).

Generate 10 to 12 competency questions covering the breadth of the domain. Avoid lookup-style questions (e.g. "what is the label of X"); favour questions that exercise structural inference, classification, or aggregation.

DOMAIN DESCRIPTION:

<paste your one-paragraph domain description here>

Begin.

Maturity model level 2: the plugin - GRL have an advanced plugin for modelling available, but for purposes of this demo we are using LLM skills

Plugin

an orchestration of skills

Workflow that fixes a problem, regenerates downstream artefacts, re-runs validation in one continuous flow.

W H E N

Repeated workflow.

E X A M P L E

GRL ontology modeller plugin.

What this gives you: agent can describe and execute

- *documented patterns*
- *domain expertise*
- *methodology*
- *interact with application systems*
- *read live data*
- *write outputs*
- *return structured results*
(rather than just reasoning about what might be there)

```
Plugins Discover Installed Marketplaces Errors

ontologist @ graph-research-labs
Scope: user
Version: 1.0.0
Modular ontology engineering plugin suite for OWL 2, SHACL, SPARQL, triple store management,
and semantic web development.

Author: Graph Research Labs
Status: Enabled

Installed components:
• Skills: fhir-to-owl, jsonld-context-design, semantic-web-developer, sparql-expertise,
owl-profile-compliance, graph-analytics, stardog-connector, ontology-modularization,
ontology-metrics-and-quality, owl-to-openapi, oxigraph-connector, rule-authoring, jena-guide,
ontology-visualization, skos-vocabulary-management, widoco-documentation, neptune-connector,
ontology-authoring-and-reviewing, shacl-validation, ontology-alignment,
data-quality-rule-generation, ontology-manager-api, sparql-optimization,
rdf-format-conversion, provenance-tracking, ontology-versioning, test-instance-generator,
graphdb-connector, shex-complementary-validation, sparql-federation, owlapi-guide,
relational-to-rdf, turtle-authoring, sparql-testing, rdflib-guide, ontology-modeller,
access-control-patterns, owl-to-data-vault, fuseki-connector, competency-question-workflow,
rdf4j-guide, csv-to-rdf, ontology-governance, reasoner-integration
```

Maturity model level 3: once you have achieved competency in LLM based modelling, add the AI harness

Harness

a governed agent

Constrains an LLM agent to operate within an ontology and SHACL declarations. The declarations are the program.

W H E N

Governance-bound autonomous operation.

E X A M P L E

Semantic Agent Harness — GRL product that provides custom governance & control on LLM's.

Agents encode skills and plug-ins into autonomous instances of an LLM.

Harnesses are needed to get full value from agents. They provide:

- **Guardrails:** agents constrained to support key standards e.g. BASEL, ITAR, FHIR, AS9100, DFARS/CMMC 2.0, DoD RAI
- **Audit:** complete, immutable audit trail from regulatory requirement through to decisions, implementation and test evidence
- **Quality:** quality validated at each stage - invalid data never enters the governed graph
- **Standards based** - support common data standards
- **Memory:** automatic agent memory per user and per project
- **Identity** management for user & agent swarms
- **No lock-in:** not tied to one agent vendor or database

When NOT to use the LLM

Six places that require more human intervention and decisions, and stronger governance



Foundational class structure (the upper-ontology level)



owl:equivalentClass and owl:equivalentProperty



Anything load-bearing for downstream reasoning



Areas of legitimate domain disagreement

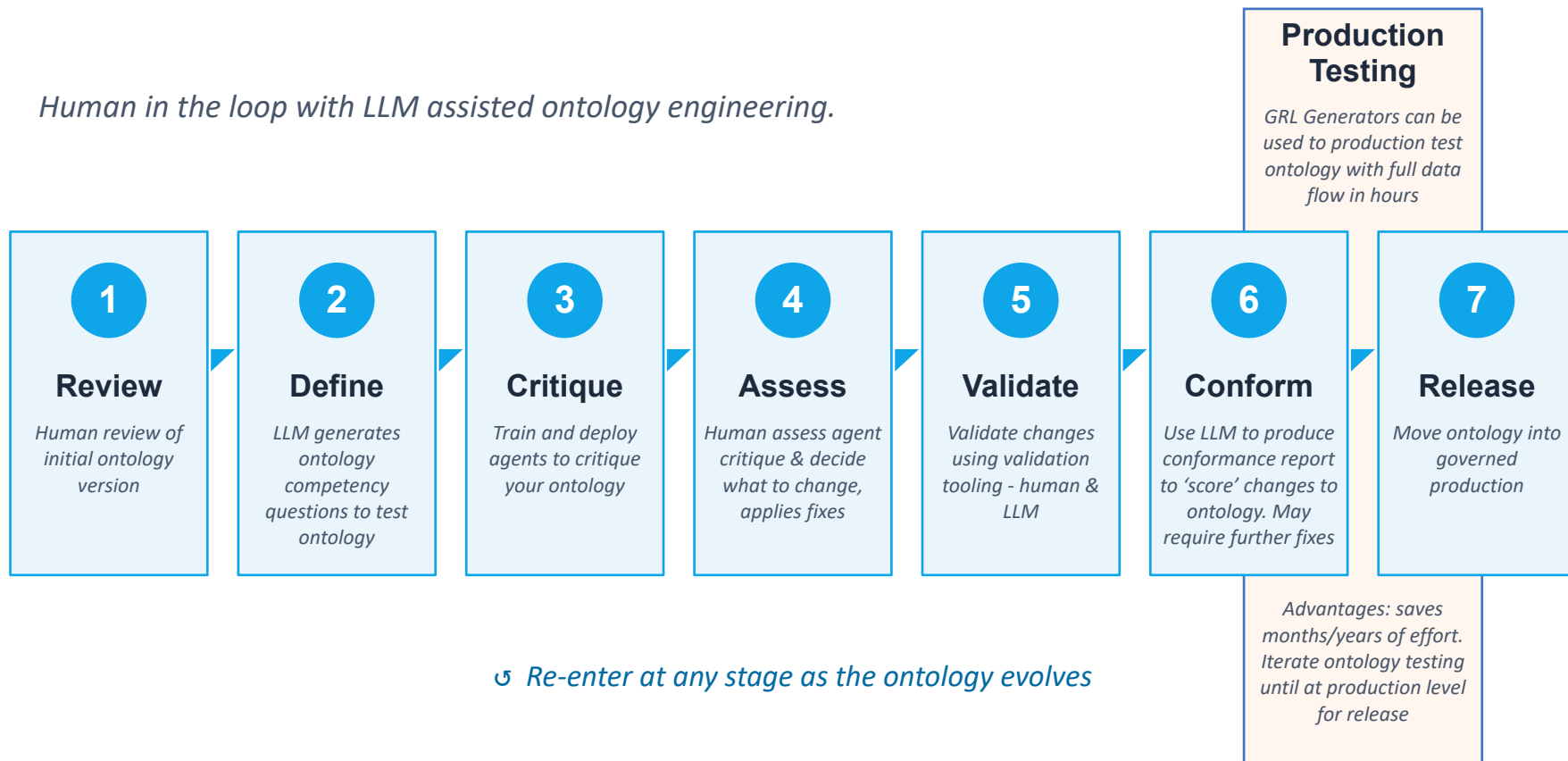


Naming of stable identifiers (IRI patterns)



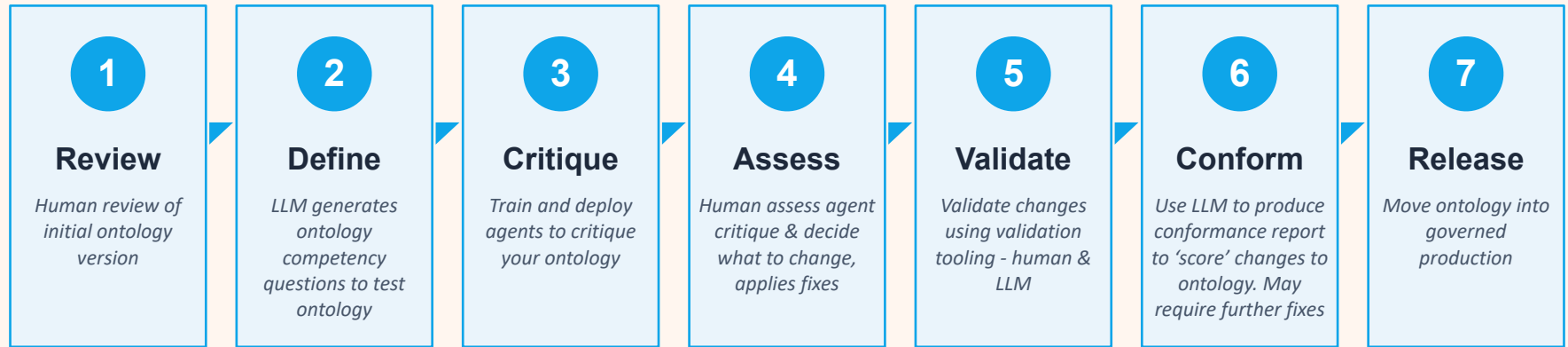
Anything safety- or compliance-critical

Human in the loop with LLM assisted ontology engineering.



Advanced methodology - Semantic Agent Harness

To improve compliance, outcome accuracy, and multi-agent coordination, you need a Semantic Agent Harness - this is an architectural pattern similar conceptually to a Firewall but with meaning - these are extremely difficult to execute reliably (>90%)



GRL Semantic Agent Harness

Advantages: governance by design, full audit trail, agent memory, standards-based guardrails (e.g. BASEL, ITAR, FHIR, AS9100, DFARS/CMMC 2.0, DODAF, DoD RAI), automatic quality management, Identity management for user & agent swarms

Graph Research Labs makes a range of tools that create and manage knowledge graphs and provide AI Safety using our Semantic Harness tool for ontology controlled agents.

Define once.
Generate forever.



Graph
Research
Labs

July 2026

Dougal Watt · Graph Research Labs

Demonstration

- Create, edit and publish ontologies using the GRL Ontology Manager tool
- Create an API, React App and MPC Server in just a few minutes using the GRL Generator tool
- Control an agents behaviour using our GRL Semantic Agent Harness tool



Graph
Research
Labs

July 2026

Dougal Watt · Graph Research Labs

1

This methodology is portable.

The skills in the handout work with any LLM.

2

The audit & conformance gap is critical.

Use validation tools to surface structural issues early on.

3

Resync on each edit/fix cycle.

Every fix triggers a SHACL update, a CQ re-run, a reasoner re-run. This discipline is what separates the methodology from ad hoc LLM use. Lint using tools such as GRL OntoLinter.

4

Test in production using tools.

Your ontology must be **tested in the real world**. Integrate with production test data and existing technology systems using tools like GRL Generators and GRL Semantic Harness. Production testing using tools is a **critical step in ontology governance**.

INFORMATION LINKS

All the information you need for this workshop is available at:

<https://graphresearchlabs.com/workshop-with-yitae-jeong-2026-links/>

You can book a free 30 minute live demonstration of GRL tools on our website: <https://graphresearchlabs.com/book-a-call/>

Connect on LinkedIn at Dougal Watt or email me at:

dougal.watt@graphresearchlabs.com



Graph
Research
Labs

July 2026

Dougal Watt · Graph Research Labs