# Why a Knowledge Graph Can Become More Complete and Less True

## Ontology Drift, Representational Hardening, and Semantic Fidelity

**Reality Drift Framework — Short Concept Note #3**

**A. Jacobs**

## The Basic Pattern

Knowledge graphs are built to make relationships visible.

They connect entities.

Define categories.

Formalize concepts.

Preserve shared meaning.

At first, the structure reflects the domain closely.

As the graph grows, it becomes more complete, more connected, and more useful.

But completeness is not the same as truth.

The core sequence is:

lived domain → formal schema → expanding graph → structural persistence → ontology drift

## A Knowledge Graph Does Not Store Reality

A knowledge graph stores a representation of reality.

An entity represents a thing.

A relationship represents a connection.

A category represents a distinction.

A property represents an attribute.

An ontology represents assumptions about what exists and how it is related.

These structures are necessary.

Without them, large knowledge systems would become impossible to search, compare, automate, or coordinate.

But the graph is not the domain.

It is a formal model of the domain.

## Why Formalization Helps

Real-world knowledge is messy.

Concepts overlap.

Categories remain ambiguous.

Relationships change.

Experts disagree.

Context alters meaning.

A knowledge graph reduces this complexity.

It gives concepts stable names.

It makes relationships explicit.

It creates rules for what belongs where.

This increases legibility.

The domain becomes easier for people and machines to navigate.

But the clarity of the representation can hide the instability of what it represents.

## The Difference Between Completeness and Fidelity

A graph becomes more complete when it contains more entities, properties, categories, and relationships.

It becomes more faithful when those structures preserve the important meaning of the domain.

These are not the same thing.

A graph can contain every required field while using the wrong distinctions.

It can represent thousands of relationships while missing the relationships that matter.

It can eliminate empty values while preserving obsolete assumptions.

Completeness measures coverage inside the model.

Fidelity measures alignment between the model and reality.

## Ontology Drift

**Ontology Drift** occurs when a formal system of categories and relationships remains stable while the domain it represents changes.

The ontology still functions.

The schema remains valid.

The graph continues expanding.

Queries still return results.

But the meaning of the entities and relationships gradually shifts.

A category may retain its name while its real-world boundaries change.

A relationship may remain encoded after it has become misleading.

A distinction that once mattered may no longer reflect how the domain actually works.

The structure survives.

The meaning moves.

## Why Ontologies Harden

An ontology often begins as a practical attempt to organize knowledge.

Over time, other systems become dependent on it.

Databases use its fields.

Search systems use its categories.

APIs use its entities.

Teams use its vocabulary.

Models train on its labels.

Reports rely on its relationships.

The ontology becomes infrastructure.

Changing it is no longer a conceptual decision alone.

It becomes a technical, organizational, and political cost.

This is **Representational Hardening**.

A provisional map becomes a durable system boundary.

## Representational Hardening

Representational Hardening occurs when a flexible interpretation becomes embedded in structures that are difficult to revise.

The sequence often looks like this:

working distinction → formal category → shared schema → system dependency → structural permanence

Once hardened, the representation gains authority.

People adapt their language to the ontology.

New data is forced into existing categories.

Exceptions are treated as data quality problems.

The domain begins conforming to the model built to describe it.

The map does not merely represent the territory.

It starts organizing the territory.

## When Classification Replaces Understanding

Classification makes knowledge manageable.

But classification can also replace direct interpretation.

A concept becomes whatever category it has been assigned.

A relationship becomes whatever edge the graph permits.

An object becomes the sum of its properties.

Ambiguity disappears from the interface even when it remains in reality.

The system becomes easier to query.

The domain becomes harder to see outside the schema.

Understanding is replaced by navigation through predefined distinctions.

## Semantic Drift Inside Stable Labels

A label can remain unchanged while its meaning changes.

A team may continue using the same category across years.

Different people begin using it differently.

New cases enter the category.

Old boundaries weaken.

The term remains stable in the graph.

Its semantic content becomes less stable in practice.

This is **Semantic Drift**.

The graph appears consistent because the label persists.

The underlying concept may no longer be shared.

## The Danger of Inherited Assumptions

Knowledge graphs often preserve assumptions longer than anyone remembers making them.

A relationship may have been added for a specific project.

A hierarchy may reflect an old organizational structure.

A category may encode one expert's interpretation.

A property may exist because an earlier system required it.

As the graph grows, these decisions become difficult to distinguish from facts about the domain.

The representation acquires the appearance of inevitability.

Historical choices become ontological claims.

## AI and Knowledge Graphs

AI systems make knowledge graphs easier to build and expand.

Models can extract entities.

Infer relationships.

Generate metadata.

Suggest categories.

Link similar concepts.

Fill missing fields.

This increases scale.

It can also accelerate drift.

The system may add plausible relationships without sufficient grounding.

Generated classifications may inherit the ontology's existing assumptions.

The graph becomes denser.

Its errors become more structurally embedded.

Automation can increase completeness faster than human judgment can preserve fidelity.

## Why More Connections Can Reduce Clarity

A larger graph often appears more intelligent.

More edges suggest more knowledge.

More connected entities suggest greater understanding.

But connection density can make weak relationships look meaningful.

Indirect associations may appear equivalent to direct ones.

Generated links may overwhelm carefully validated ones.

The graph becomes richly connected but harder to interpret.

More structure produces more possible paths.

It does not necessarily produce better understanding.

## Semantic Fidelity

**Semantic Fidelity** describes how well a representation preserves the important meaning, distinctions, context, and relationships of what it represents.

In a knowledge graph, high semantic fidelity requires more than correct syntax.

The entities must correspond to meaningful things.

The relationships must preserve the nature of the connection.

The categories must reflect useful distinctions.

The graph must remain revisable when the domain changes.

A technically valid graph can have low semantic fidelity.

A sparse graph can have high semantic fidelity if its structure preserves what matters.

## Why the System Still Looks Healthy

Ontology Drift is difficult to detect because visible indicators continue improving.

The entity count rises.

The graph becomes denser.

Coverage increases.

The schema becomes more complete.

Search returns more results.

Fewer fields remain empty.

These are measures of representational expansion.

They do not show whether the representation remains answerable to the domain.

The graph looks healthy because its internal structure becomes richer.

The loss of external alignment remains harder to measure.

## A Recognition Test

Ask:

Do the categories still reflect how experts currently understand the domain?

Can the graph represent exceptions without distorting them?

Are relationships being validated, or merely inferred and retained?

Can anyone explain why the major categories and edges exist?

Do users adapt their understanding to the graph because changing the graph is too difficult?

Would the same ontology be designed today from direct observation of the domain?

These questions test whether the graph remains a living representation or has become hardened infrastructure.

## Knowledge Graphs and Reality Drift

Within the **Reality Drift framework**, a knowledge graph is a scaled symbolic system.

It allows people and machines to navigate a domain through entities, categories, properties, and relationships.

As the graph becomes more important, direct contact with the domain may become less influential.

Reality Drift begins when the graph remains coherent, complete, and operational while becoming less answerable to the reality it was built to organize.

The schema still validates.

The queries still run.

The meaning has changed elsewhere.

## The Core Principle

A knowledge graph becomes more complete by adding structure.

It becomes more true by remaining corrigible.

Those processes can move in opposite directions.

**A knowledge graph can become more complete and less true when internal coverage expands faster than external correction.**

**Ontology Drift begins when the structure remains stable but the domain no longer fits the distinctions embedded inside it.**

**Representational Hardening turns yesterday's interpretation into today's infrastructure.**

**Semantic Fidelity depends not on how much the graph contains, but on whether its categories and relationships remain answerable to reality.**

**Keywords:** knowledge graph problems, ontology drift, why knowledge graphs fail, knowledge graph accuracy, knowledge graph completeness, ontology engineering, taxonomy drift, schema drift, semantic drift, semantic fidelity, representational hardening, knowledge representation, enterprise knowledge graph, graph database, entity relationships, metadata management, semantic search, information architecture, AI knowledge graphs, automated entity extraction, ontology maintenance, knowledge graph governance, Reality Drift, A. Jacobs