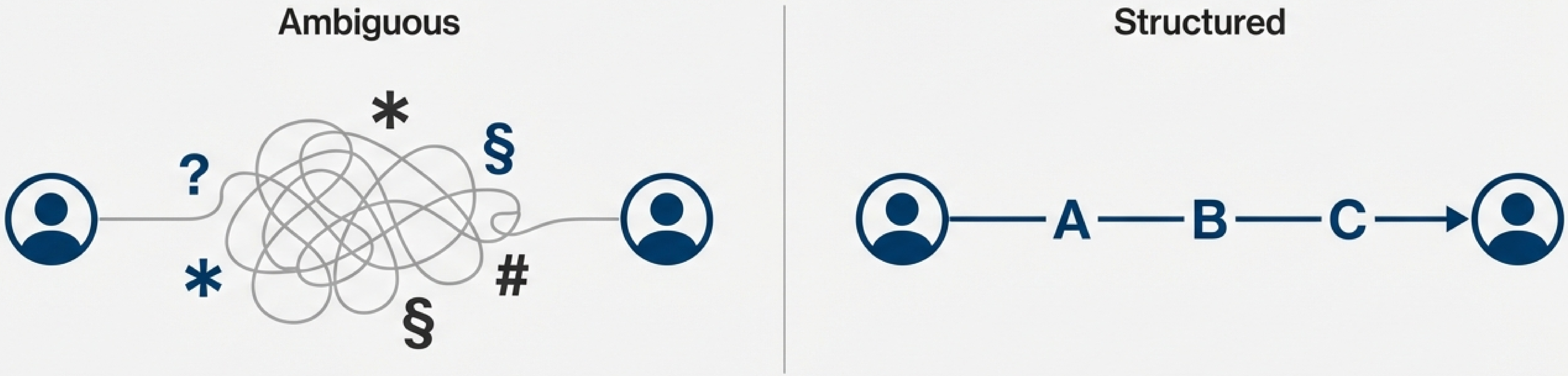
Taming Ambiguity

How Ontologies Create Shared Meaning for Intelligent Systems

Based on “Ontology and Information Systems,” Stanford Encyclopedia of Philosophy.

Communication relies on a significant body of shared understanding.



When any two agents communicate, they must agree on the meaning of the symbols they use. Humans have long used dictionaries to codify these shared meanings.

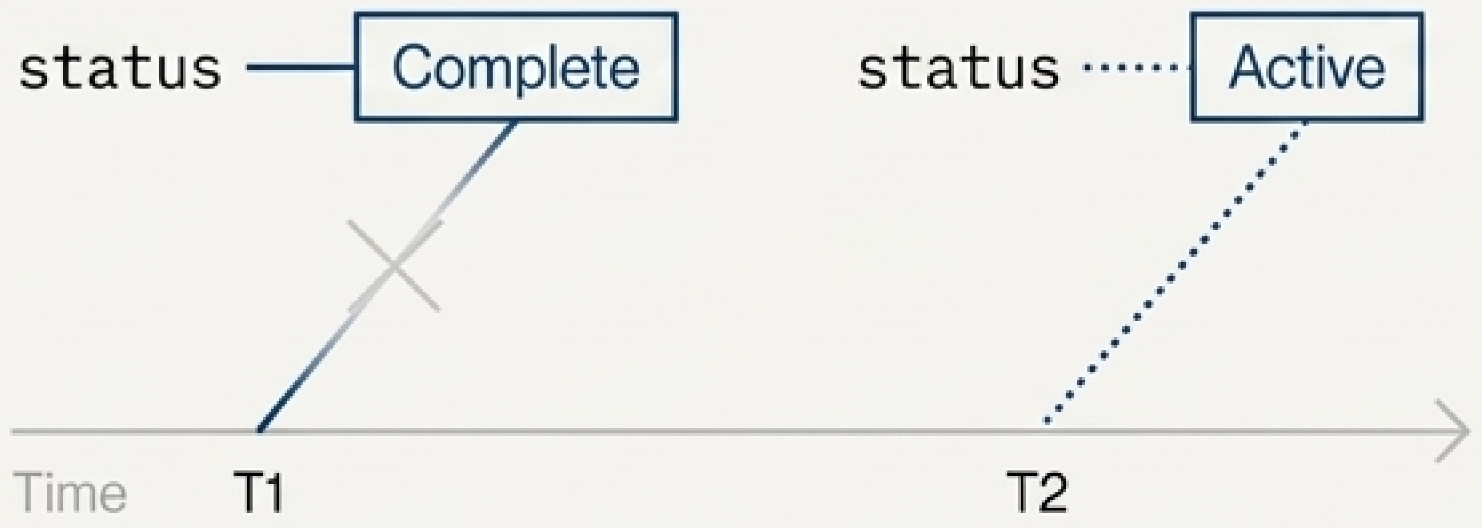
For machines, this challenge is amplified. They cannot rely on experience or intuition. Meaning must be stated with mathematical precision.

Ontology in information systems is the discipline that attempts to create these shared meanings for computation.

In software, a lack of shared meaning leads to critical failures.

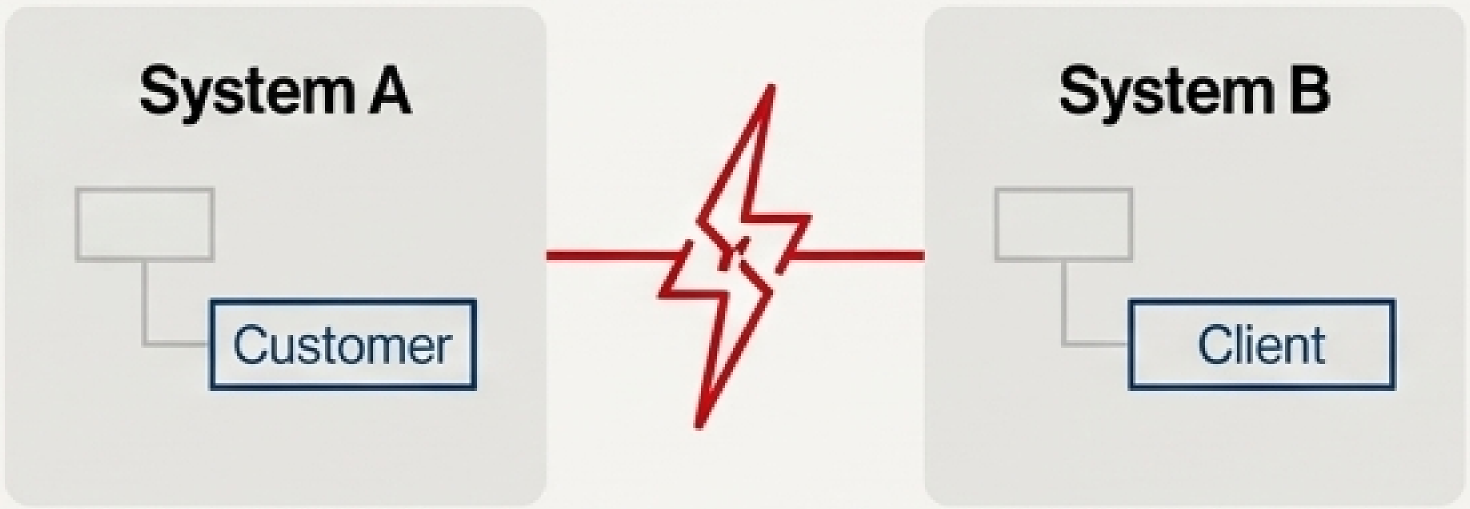
Concept Drift

The meaning of symbols used in code by different programmers changes over time, causing unintended and unpredictable system performance.



Interoperability Failure

Systems built on different assumptions about data cannot communicate effectively, hindering integration and creating data silos.



The Goal: An ontology establishes a **standard catalogue of concepts** to **prevent concept drift** and **support robust system interoperability.**

The solution is a formal ‘specification of a conceptualization’.

> “ a specification of a conceptualization ”
>
> — Thomas Gruber, 1993

A **conceptualization** is a way of understanding the **structure of a concept**, its **relationship to other concepts**, and the **principles** that govern it.

A **specification** is the recording of that understanding in a formal way, removing reliance on human intuition.

Philosophy

Seeks to identify the fundamental constituents of *reality*.
“What ultimately exists?”

Computer Science

Seeks to identify what is *assumed to exist* in a domain for a specific computational purpose.
“What is useful for this model?”

Ontologies exist on a spectrum of formal expressiveness.

Low Formality (Glossary)

Collections of words with natural language definitions. Relies on human interpretation.

Structural (Taxonomy / Graph)

Defines parent-child relationships (is a) or simple binary relations between concepts (e.g., Knowledge Graphs in RDF).

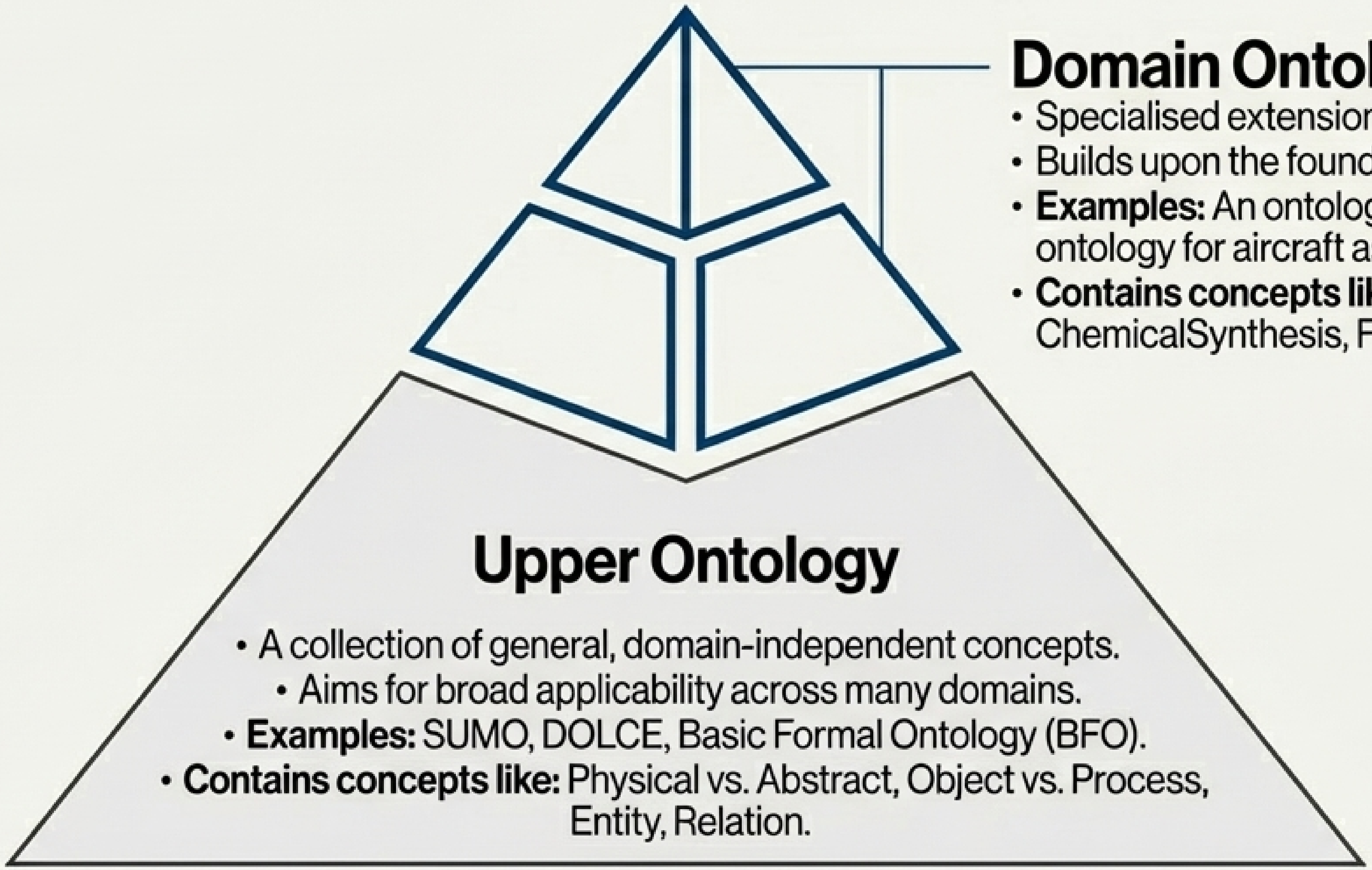
Formal (Description Logic)

A subset of first-order logic with decidable inference. The foundation for languages like OWL (Ontology Web Language).

Highly Formal (First-Order & Higher-Order Logic)

Extremely expressive mathematical logics that allow for complex, generalised statements about the world.

Knowledge is layered: Upper Ontologies provide the foundation for Domain Ontologies.



Domain Ontologies

- Specialised extensions for specific applications.
- Builds upon the foundation of an upper ontology.
- **Examples:** An ontology for drug chemistry, an ontology for aircraft assembly.
- **Contains concepts like:** DrugInteraction, ChemicalSynthesis, Fuselage.

The core elements of an ontology define what is, what happens, and how things relate.

Common Categories (The Nouns)

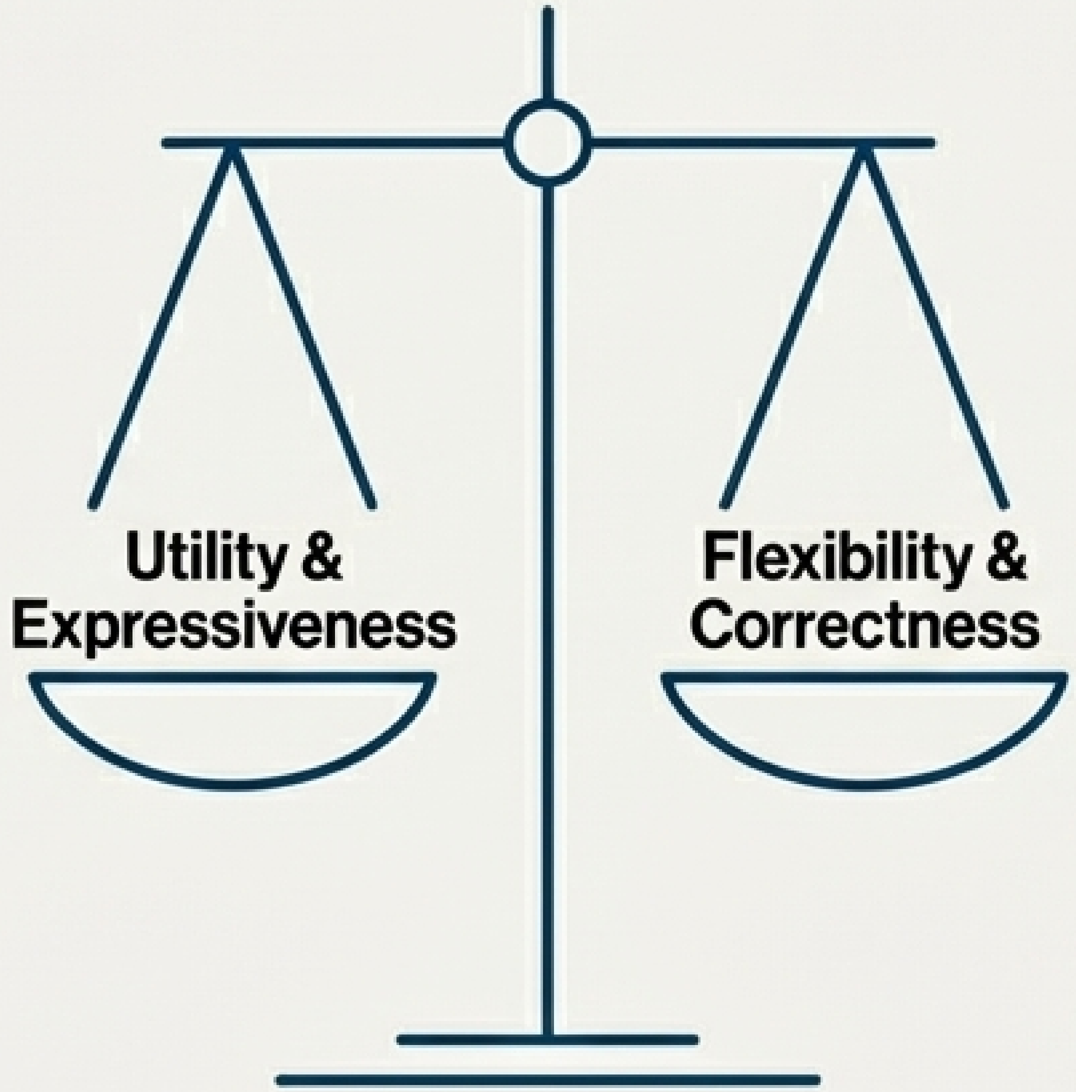
- **Thing/Entity:** The class of all things.
- **Physical:** Things with a position in space and time.
    - **Object/PhysicalThing:** Tangible entities that *are*.
    - **Process/Event:** Things that *happen*.
- **Abstract:** Things without a position in space and time (e.g., numbers, concepts).
- **Relation**: Relationships among entities.

Common Relationships (The Verbs)

- `subclass of` ('is a'): A class is more specific than another.
- `instance of`: An individual is a member of a class.
- `physical part of` ('has a'): Mereological relation (e.g., a wheel is part of a car).
- `temporal part of`: An event is part of a larger process.
- **Temporal Relations:** `before`, `meets`, `during` (Allen, 1984).
- **Case Roles:** `agent`, `patient`, `instrument`.

Ontological commitment is a balancing act between utility and correctness.

- **Saying too little (Error of Omission):** An ontology with too few concepts or definitions has limited utility. One can avoid being wrong by saying nothing.
- **Saying too much (Error of Commission):** An ontology making false or overly rigid claims (e.g., "the Earth is flat") excludes useful models of the world.



Example: Strong vs. Weak Commitment

Strong Commitment (Unwise):

`∃x unicorn(x)`

This asserts that a unicorn exists, a commitment that is likely false.

Weak Commitment (Safe & Useful):

```
∀x (unicorn(x) ⇒
∃y (horn(y) ∧ part(y,x)))
```

This only commits to the fact that if a unicorn were to exist, it would have a horn as a part. This defines a concept without asserting its existence.

The choice of logic dictates the trade-off between expressiveness and decidability

Description Logics (e.g., OWL)	First-Order & Higher-Order Logic (e.g., KIF, SUMO)
Expressiveness A subset of First-Order Logic. Primarily for classification and type reasoning. Limited to binary relations.	Expressiveness Far greater. Can state general rules (`∀x Organism(x) ⇒ Mortal(x)`), quantify over relations (in HOL), and define concepts like transitivity axiomatically.
Computation **Decidable.** Inference is guaranteed to terminate. Queries will always return an answer (or confirm no answer exists) in finite time.	Computation **Semi-decidable (FOL).** An automated theorem-prover is guaranteed to find a proof if one exists, but may not terminate if one does not.
Use Case The foundation of the Semantic Web; valued for its computational guarantees.	Use Case Large-scale knowledge bases (e.g., Cyc); valued for capturing complex world knowledge.

The quest to categorise knowledge has a long and varied history.

Ancient Roots
Porphyry's Tree (3rd century) – using a tree structure to show concept relationships.

Scientific Cataloguing
Linnaeus (1758), Wilkins (1685) – systematic efforts to define categories in natural science and language.

Modern Logic
Frege & Peirce – laying the foundations for formal, mathematical logic.

Early AI
Expert Systems (MYCIN, 1970s), Semantic Networks – early computational use of structured knowledge.

Large-Scale Efforts
The Cyc Project (1989) – a massive repository of common-sense knowledge.

A Recurring Debate: “Neats” vs. “Scruffies”

A historical tension in AI development.

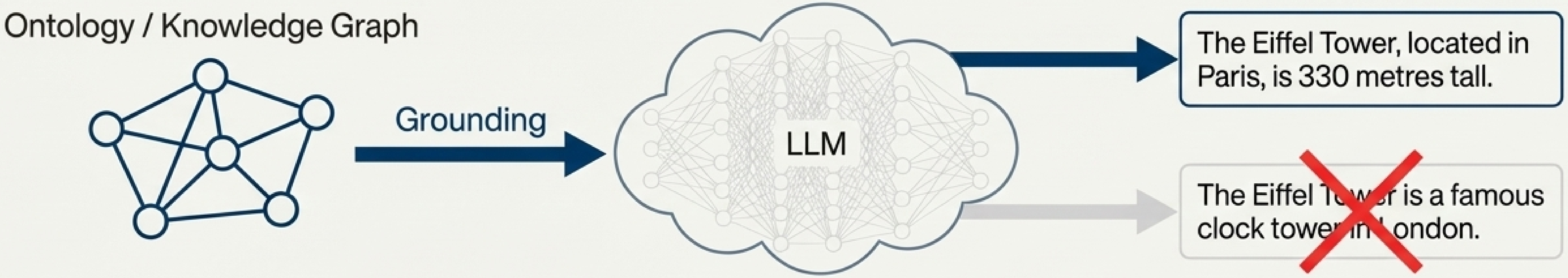
Neats: Argue that knowledge representation requires a formal mathematical basis and theoretical rigour.

Scruffies: Argue that a system that "just works" is sufficient, even without a complete theoretical underpinning.

How to assess the quality and rigour of an ontology

- ☑ **Logical Foundation:** Is a mathematical logic used? If so, which one (Description Logic, FOL, HOL)? This dictates what can be formally expressed.
- ☑ **Computational Validation:** Has the ontology been checked for consistency with an automated reasoner (e.g., FaCT++, Pellet, Vampire)?
- ☑ **Scale & Scope:** What is the number of concepts (classes, instances) and relationships defined?
- ☑ **Axiomatic Depth:** How many formal axioms or formulas are there? What is their logical complexity (e.g., arity of relations, use of quantifiers)?
- ☑ **Richness:** Does it support numbers and arithmetic?
- ☑ **Lexical Grounding:** Are there mappings to lexical resources like WordNet, and in how many languages?

Ontologies provide the world model that Large Language Models lack.



The Problem

LLMs are models of **language**, not models of the **world**.

This can lead to "hallucinations" – generated text that is fluent but factually incorrect or logically inconsistent with reality.

The Neuro-Symbolic Approach

Combining neural networks (like LLMs) with symbolic reasoning systems (like ontologies).

Ontologies, often as Knowledge Graphs, can be used to ground LLMs in factual, structured knowledge.

This helps constrain LLM output to conform to known facts, taxonomic relationships, and complex logical rules about the world.

Ontologies are actively used to structure knowledge in critical domains.

Biology & Life Sciences

Structuring vast amounts of genomic and biomedical data (e.g., Gene Ontology). Standardising vocabulary for research.

Engineering & Manufacturing

Knowledge management for complex system design, such as aircraft assembly. Ensuring consistency in parts and processes.

Internet of Things (IoT)

Enabling interoperability between billions of diverse sensors and devices, allowing them to share data meaningfully.

Natural Language Processing (NLP)

Providing a semantic framework (e.g., an ontology of space) to allow systems to understand the meaning behind text, not just the words.

The future lies in deeper formalisms and more rigorous evaluation.

Future Directions

Computational Metaphysics: A growing collaboration between computer science and philosophy to formally axiomatise fundamental aspects of reality (e.g., theories of parts, places, substances).

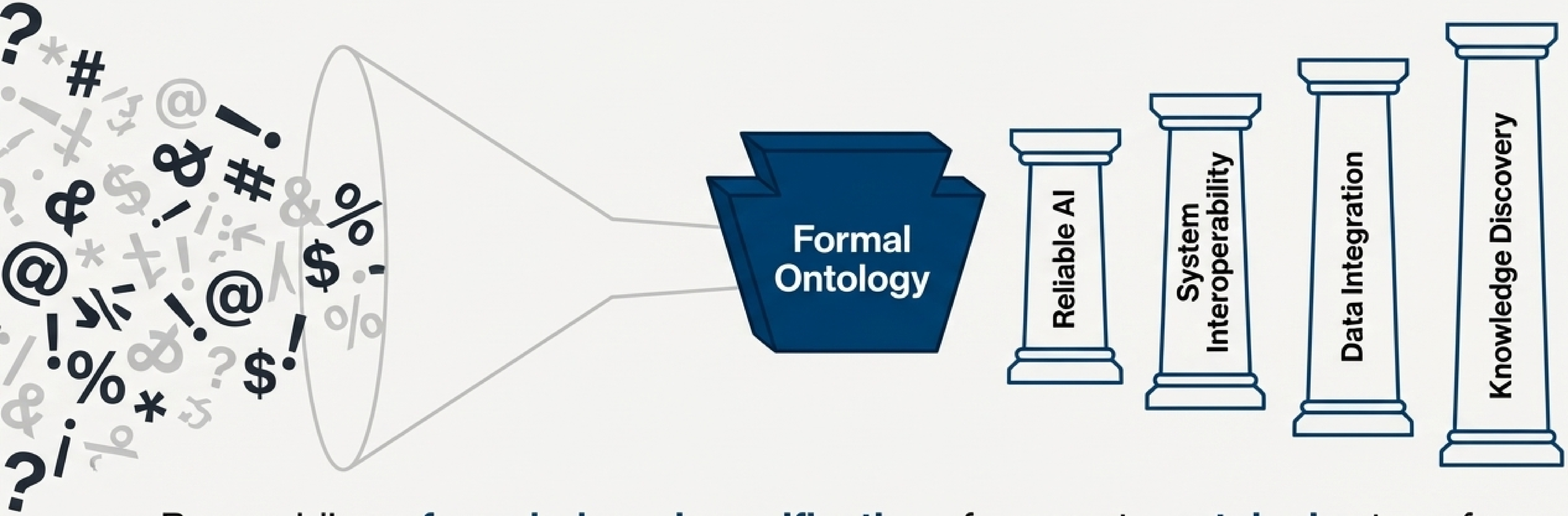
More Expressive, Performant Logics: The automated theorem-proving community continues to advance reasoners that can handle larger, more complex theories, making expressive logics more practical.

Competency Questions: A systematic approach to evaluating ontologies by defining a corpus of questions a knowledgeable system should be able to answer, moving beyond purely structural metrics.

An Open Question

> "When does a part of a cake stop being a cake and start to be a sugar molecule? Formalising these limits of granularity remains a key challenge."

From Ambiguity to Intelligence



By providing a **formal**, **shared specification** of concepts, **ontologies** transform chaotic, ambiguous information into a **structured foundation**. This keystone of meaning enables robust, **interoperable**, and truly **intelligent** systems.