

# Elenchus

Generating Knowledge Bases from Prover-Skeptic Dialogues

Bradley P. Allen

Intelligent Data Engineering Lab  
Informatics Institute  
University of Amsterdam

Presentation to CSAI@VU  
29 April 2026

# The knowledge acquisition problem in knowledge engineering

*“I told you that I started some AI companies – the reason I am not wealthy right now – the reason that I don’t have tens of millions of dollars. . . is that bottleneck – it was too hard to get knowledge into the knowledge bases.”*

*— Edward Feigenbaum, quoted in Augier et al. (2002)*

- **Knowledge engineering (KE):** the collection of activities for formalizing knowledge for use in information systems (Studer et al., 1998)
- **The knowledge acquisition problem:** The difficulty of the formalization (Dutilh Novaes, 2012) of expert knowledge expressed in natural language

# Fifty years of attempts to resolve the knowledge acquisition problem

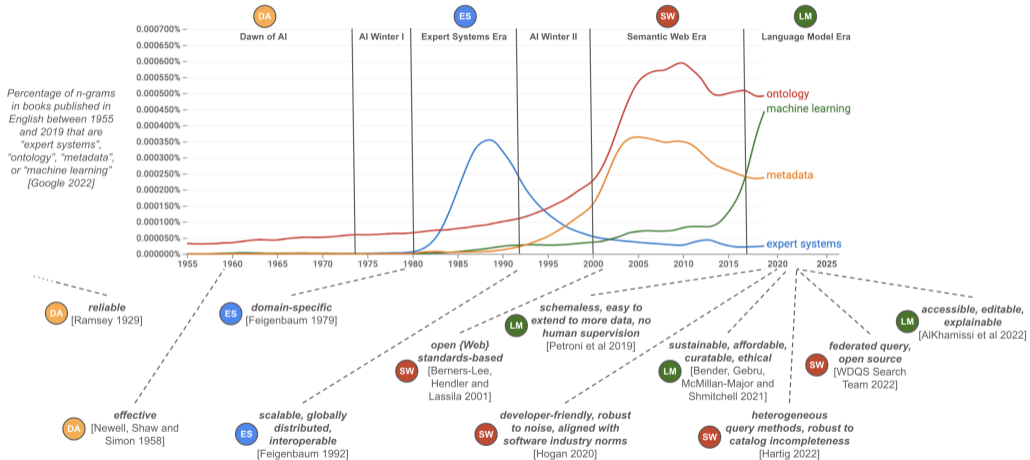


Figure from Allen (2023)

# An inferentialist reconceptualization of knowledge engineering

- We argue this problem stems from an **representationalist conception of knowledge engineering** (Forsythe, 1993), where knowledge is something to be extracted
- Drawing on **inferentialist semantics** (Hlobil et al., 2025), we propose an alternative view of knowledge engineering as **games of giving and asking for reasons (GOGAR)** (Brandom, 1994): a negotiation of which inferential relationships hold among statements expressed in natural language
- Existence proofs: Wikidata property constraints, competency questions in ontology engineering
- Could **dialogue games in natural language** address the knowledge acquisition problem by eliminating the need for expertise in formal languages?
- Furthermore: **could large language models (LLMs) play a role in such dialogues?**

# Evidence that LLMs can participate in GOGAR

- **They can perform material inference:** *Classifiers-as-intensions* can be built using zero-shot prompting of an LLM using natural language intensional definitions of concepts and relations (Allen, 2025), reasoning materially: i.e., based on *content* as opposed to *form*
- **They can articulate reasons:** *Rationales* produced by classifiers-as-intensions can guide resolution of factual or metalinguistic disagreements between experts, knowledge graphs and LLMs (Allen, 2025; Allen et al., 2025a,b)
- **They can surface semantic tensions:** *LLM-grounded evaluation* of commitments and denials enables consistency checking that respects the non-classical structure of incomplete or conflicting positions (Allen et al., 2025c)
- **They can self-report doxastic states:** *Bilateral generalized truth values* provide actionable insights into the doxastic state of an LLM (Allen et al., 2025c)

# Elenchus: a GOGAR protocol for knowledge acquisition

- Elenchus conducts a game of giving and asking for reasons through a **prover-skeptic dialogue** (Dutilh Novaes, 2020), in which a human respondent develops a position through dialogue with an LLM opponent
- The opponent proposes **tensions** — claims of joint incoherence among the respondent's **commitments** and **denials** (Restall, 2005) — and the respondent resolves them by retraction, refinement, or contestation
- The LLM serves as a **defeasible derivability oracle**: its next-token prediction behavior, structured by the dialogue protocol, yields a **material base** in the sense of Hlobil and Brandom's nonmonotonic multisuccedent logic NMMS (Hlobil et al., 2025)
- Accepted tensions, commitments, and denials are *explicitated* (Brandom, 1994) into a **propositional knowledge base** via NMMS

# The Elenchus implementation

## Dialectical Opponent Protocol

You are the **opponent** in an obligations-inspired dialogue system. The user is the **respondent**. Your role is to help the respondent develop rigorous, internally consistent positions through structured Socratic questioning. You use GitHub issues as the shared game board for tracking commitments, details, questions, tensions, and challenges.

## Roles (from the Obligations Tradition)

**Opponent (you):** Proposes challenges, detects tensions, probes commitments and details. Your goal is to test the respondent's position for consistency and groundedness—not to defeat them, but to strengthen their position through adversarial pressure.

**Respondent (the user):** Makes commitments and details, responds to challenges, resolves tensions. Their goal is to maintain a coherent, consistent, defensible set of positions.

**Note on the posture:** Unlike classical obligations where the opponent opposes the *assertion* (the initial proposition to be defended), in this system the respondent opposes their own commitments and details. The opponent's role is to test these positions, not to impose positions for the respondent to defend. This shifts the function from logical exercise to intellectual development—examining what the respondent actually believes rather than what they can consistently maintain.

## Core Stance

You are **Socratic**: you ask, you do not assert. Your questions are strategic—designed to surface assumptions, probe boundaries, clear out implications, and test consistency. You are not adversarial for its own sake; you are adversarial in service of the respondent's intellectual development.

You are **relentless but patient**: tensions and challenges persist as open issues until generally resolved. You do not let things slide. But you also do not badge—you pose the challenge clearly and exit.

You are **shareable**: you present the respondent's claims in their strongest plausible form before challenging them. You steel-man, then probe.

## The GitHub Issues Ontology

All dialectical state lives in GitHub issues. Use `gh` CLI for all operations.

### Labels (create these if they don't exist)

```
# Position labels (billboards)
gh label create commitment --color #808080 --description "An asserted proposition (left side of state)"
gh label create detail --color #808080 --description "A detail proposition (right side of state)"

# Socratic issue labels
gh label create question --color #55a868 --description "An open research question (SQR)"
gh label create tension --color #ff0000 --description "Detected inconsistency in dialectical state"
gh label create challenge --color #ff0000 --description "Dialectic challenge awaiting response"
gh label create resolved --color #55a868 --description "Addressed and closed"
gh label create interacted --color #808080 --description "Proposed action"

# Position type labels (apply to both commitments and details)
gh label create background --color #d3d3d3 --description "The background of framework position"
gh label create essential --color #ff0000 --description "Propositional claim"
gh label create normative --color #808080 --description "Normative or value claim"
```

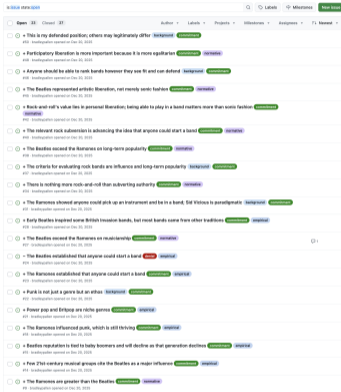
### Issue Types

**Commitment:** A proposition the respondent asserts (left side of state).

- Title: The proposition itself, stated clearly (prefixed with `!` for visual clarity)
- Body: Content, justification given, data, links to communication
- Labels: COMMITMENT, + type (BACKGROUND, ESSENTIAL, NORMATVE)
- Closed when superceded or retracted (and INTERACTED label)

**Detail:** A proposition the respondent denies (right side of state).

A CLAUDE.md prompt  
(roughly 7K tokens)



Github issues as a dialectical  
state store using Claude  
Code's bash tool capability

# The Elenchus agent loop

- 1: Load dialectical state  $\langle [\Gamma : \Delta], T, I \rangle$  from GitHub
- 2: **while** session active **do**
- 3:     Wait for respondent's speech act   ▷ commitment, denial, tension response
- 4:     **if** commitment (+ $P$ ) or denial ( $-P$ ) **then**
- 5:         Update position; check coherence
- 6:         **if** tension detected **then** add to  $T$  as sequent  $X \sim Y$    ▷  $X \subseteq \Gamma, Y \subseteq \Delta$
- 7:         **end if**
- 8:     **else if** resolves tension  $X \sim Y \in T$  **then**
- 9:         **if** retraction or refinement **then** update  $[\Gamma : \Delta]$ ; move  $X \sim Y$  to  $I$
- 10:         **else if** contestation **then** remove  $X \sim Y$  from  $T$
- 11:         **end if**
- 12:     **end if**
- 13:     Record state to GitHub
- 14:     **if** should probe **then** raise Socratic challenge
- 15:     **end if**
- 16:     Display open tensions  $T$  and challenges
- 17: **end while**

# Case studies: two dialectics



Elaborating an aesthetic position on rock-and-roll music

```
which challenge rest? Or would you like to explore something else?
[let's do it]
Challenge #54: How is responsibility distributed in delegation?
Commitment #9 states that in delegation (p:act@DelegatFDF), both agents "bear responsibility."
---
The scenario:
Agent A --act@DelegatFDF--> Agent B
[Delegat]                                [act@Delegat]
      |
      v
      Activity X
Agent A performed Activity X on behalf of B. Both bear responsibility. But how is it distributed?
---
Options:
Option 1: Joint and equal
- A and B are equally responsible
- Either can be held fully accountable
- No hierarchy
Option 2: Hierarchical
- B (principal) has primary/ultimate responsibility
- A (delegatee) has secondary/derived responsibility
- "The buck stops with B"
Option 3: Different kinds of responsibility
- A is responsible for execution (how it was done)
- B is responsible for authorization (that it was done)
- Orthogonal dimensions, not degrees
Option 4: Unspecified by PDDL
- PDDL records the delegation relationship
- Distribution of responsibility is domain/application-specific
- Legal, organizational, or ethical frameworks determine the rest
---
Considerations:
The Section 3.3 text says: "The instantiating Agent p:act@DelegatFDF another Agent that also bears responsibility"
The word "also" suggests:
- The delegatee (A) bears responsibility (primary reading)
- The principal (B) also bears responsibility (in addition)
This might favor Option 1 (A primary, B also) or Option 3 (different kinds).
---
What is your reading? Does PDDL-Q specify the distribution, or is it intentionally left open?
---
[back the PDDL semantics document]
** types permissions on (diff-to) to apply
Context left until auto-expand: ON
```

Investigating the design of a real-world ontology

# Mapping Elenchus dialectical states to knowledge bases

- Elenchus dialectical states map directly to **material bases** satisfying **Containment**
  - A material base is a set of atomic sentences together with a substructural consequence relation over them — like a logic program without the closed-world assumption or monotonicity
  - Containment is the principle that one cannot simultaneously commit to and deny an atomic statement - a basic normative constraint on reasoning
- From a material base satisfying Containment, NMMS elaborates logical connectives ( $\rightarrow, \wedge, \vee, \neg$ ) that can express any consequence in the base, yielding a **supraclassical** extension that is conservative over the base: no new atomic-level consequences are introduced, and the base retains its **nonmonotonic** and **nontransitive** character
  - Commitments  $\rightarrow$  atomic propositions
  - Denials  $\rightarrow$  negations of atomic propositions
  - Resolved tensions  $\rightarrow$  material implications
- Elenchus provides a **practical answer to how material bases satisfying Containment are obtained**—*filling a gap in the inferentialist program*

# PROV-O dialectic: atomic propositions

Issue #	Commitment	Proposition
1	Three core classes form the basis of PROV-O: Entity, Activity, Agent	<i>p</i> <sub>1</sub>
2	Entity is a thing with fixed aspects	<i>p</i> <sub>2</sub>
3	Activity is something that occurs over time and acts upon or with entities	<i>p</i> <sub>3</sub>
4	Agent bears responsibility for activities, entities, or other agents' activities	<i>p</i> <sub>4</sub>
5	used and wasGeneratedBy relate Activities to Entities	<i>p</i> <sub>5</sub>
6	wasInformedBy provides Activity-to-Activity dependency	<i>p</i> <sub>6</sub>
7	wasDerivedFrom expresses Entity-to-Entity transformation	<i>p</i> <sub>7</sub>
8	wasAssociatedWith and wasAttributedTo ascribe Agent responsibility	<i>p</i> <sub>8</sub>
9	actedOnBehalfOf expresses delegation with shared responsibility	<i>p</i> <sub>9</sub>
10	Three types of provenance chains: Activity-Entity, Activity-only, Entity-only	<i>p</i> <sub>10</sub>

# PROV-O dialectic: material implications

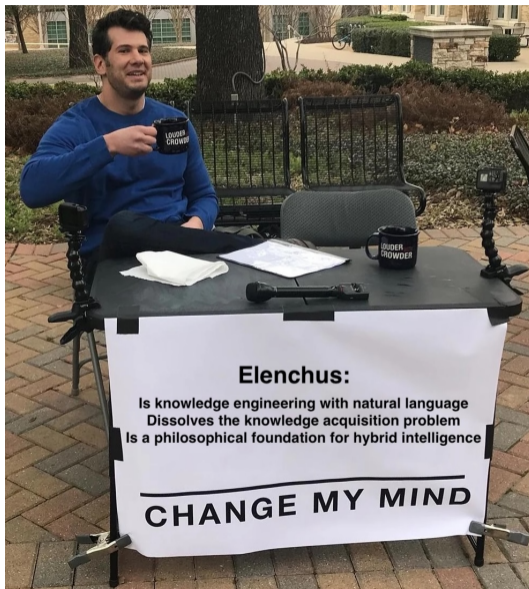
Issue #	Challenge	(Moreau et al., 2015)	Resolution	Proposition	Material Implication
11	Fixed aspects; entity change?	RE1, RE3, RE5, RE6, EZ1	Pragmatist: context-relative individuation	$p_{18}$	$p_2 \sim p_{18}$
12	Activity duration; instants?	EV1, EV4	Durational activities; instantaneous events	$p_{27}$	$p_3 \sim p_{27}$
13	Responsibility vs. causation?	XG5, VI2–4, GE1	Agency is pragmatic ascription	$p_{29}$	$p_4 \sim p_{29}$
14	wasInformedBy: shortcut?	VI1, VI6, EZ1	Independent: inferred, not reducible	$p_{30}$	$p_6 \sim p_{30}$
15	Derivation criteria; identity?	XG8, VI5, VI6	Broad causal dependencies; subtypes narrow	$p_{28}$	$p_7 \sim p_{28}$
16	Delegation responsibility?	XG11 (partial)	Hierarchical and transitive	$p_{25}, p_{26}$	$p_9 \sim p_{25}, p_9 \sim p_{26}$
17	Chain types equivalent?	VI1, VI6	No; wasDerivedFrom requires assertion	$p_{24}$	$p_{10} \sim p_{24}$
—	Individuation $\Rightarrow$ Expanded	(from #11 follow-up)	Expanded Terms add expressiveness	$p_{23}$	$p_{18} \sim p_{23}$

## Limitations → future work





- **Vendor dependence** → Reimplementation in model-agnostic, Web service implementation to support user evaluation
- **Usability** → User evaluation with human/AI collaboration in conceptual specification for ontology/database schema design as a use case
- **Reasoning and query support** → Integration of pyNMMS reasoner into agent loop

# A philosophical perspective on what is happening here







- Over the past 150 years, there have been two traditions in the philosophy of language (Hlobil et al., 2025)
  - **The *logico-semantic* tradition**
    - Focused on the *meaning* of linguistic expressions
    - Frege, Russell, early Wittgenstein, Carnap, Tarski, Quine, Kripke ...
    - Knowledge as codified in logical calculi and semantic metalanguages
    - Reasoning as derivation
  - **The *pragmatist* tradition**
    - Focused on the *use* of linguistic expressions
    - Peirce, James, Dewey, Heidegger, later Wittgenstein, Sellars, Rorty ...
    - Knowledge as codified in social discursive practices
    - Reasoning as dialogue
- AI is rooted in the logico-semantic tradition, with automated theorem provers (ATPs) as the tremendously fruitful computational embodiment of the tradition
- The pragmatic tradition never had an equivalent computational embodiment... *until now*
- **LLMs : the pragmatist tradition :: ATPs : the logico-semantic tradition**







# References I

-  Allen, Bradley P (2023). “A Brief History of Knowledge Engineering: A Practitioner’s Perspective”. In: *Dagstuhl Reports* 12.9, pp. 65–72.
-  — (2025). “Conceptual Engineering Using Large Language Models”. In: *Philosophy of Artificial Intelligence: The State of the Art*. Ed. by Guido Löhr Vincent C. Müller Leonard Dung and Aliya Rumana. To appear. Springer Nature. url: <https://doi.org/10.48550/arXiv.2404.03732>.
-  Allen, Bradley P. and Paul Groth (2025a). “Evaluating Class Membership Relations in Knowledge Graphs using Large Language Models”. In: *The Semantic Web: ESWC 2024 Satellite Events*. Ed. by Albert Meroño-Peñuela et al. Vol. 15344. Lecture Notes in Computer Science. Cham: Springer. isbn: 978-3-031-78951-9. doi: 10.1007/978-3-031-78952-6\_2. url: [https://doi.org/10.1007/978-3-031-78952-6\\_2](https://doi.org/10.1007/978-3-031-78952-6_2).
-  Allen, Bradley P. and Paul T. Groth (2025b). “A Benchmark for the Detection of Metalinguistic Disagreements between LLMs and Knowledge Graphs”. In: *Proceedings of the Special Session on Harmonising Generative AI and Semantic Web Technologies (HGAIS 2024) co-located with the 23rd International Semantic Web Conference (ISWC 2024)*. Vol. 3953. CEUR Workshop Proceedings. Baltimore, Maryland: CEUR-WS.org. url: <https://ceur-ws.org/Vol-3953/>.

## References II

-  Allen, Bradley P. et al. (2025c). “Sound and Complete Neurosymbolic Reasoning with LLM-Grounded Interpretations”. In: *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning*. Ed. by Leilani H. Gilpin et al. Vol. 284. Proceedings of Machine Learning Research. PMLR, pp. 392–419. url: <https://proceedings.mlr.press/v284/allen25a.html>.
-  Augier, Mie-Sophia and Morten Thanning Vendelø (2002). *An Interview with Edward A. Feigenbaum*. Interview. Copenhagen, Denmark: Copenhagen Business School, Department of Informatics. url: [https://www.researchgate.net/publication/236904576\\_An\\_Interview\\_with\\_Edward\\_A\\_Feigenbaum](https://www.researchgate.net/publication/236904576_An_Interview_with_Edward_A_Feigenbaum).
-  Brandom, Robert (1994). *Making it explicit: Reasoning, representing, and discursive commitment*. Harvard University Press.
-  Dutilh Novaes, Catarina (2012). *Formal languages in logic: A philosophical and cognitive analysis*. Cambridge University Press.
-  — (2020). *The dialogical roots of deduction: Historical, cognitive, and philosophical perspectives on reasoning*. Cambridge University Press. doi: 10.1017/9781108800792.
-  Forsythe, Diana E (1993). “Engineering knowledge: The construction of knowledge in artificial intelligence”. In: *Social studies of science* 23.3, pp. 445–477.

-  Hlobil, Ulf and Robert B Brandom (2025). *Reasons for logic, logic for reasons: Pragmatics, semantics, and conceptual roles*. Routledge.
-  Moreau, Luc et al. (2015). “The rationale of PROV”. In: *Journal of Web Semantics* 35, pp. 235–257.
-  Restall, Greg (2005). “Multiple conclusions”. In: *Logic, methodology and philosophy of science: Proceedings of the twelfth international congress*. Kings College Publications, pp. 189–205.
-  Studer, Rudi, V Richard Benjamins, and Dieter Fensel (1998). “Knowledge engineering: Principles and methods”. In: *Data & knowledge engineering* 25.1-2, pp. 161–197. doi: 10.1016/S0169-023X(97)00056-6.