AI-Assisted Modeling in Economics: Transparency and Verification

Special Issue Proposal

Christopher Carroll and Akshay Shanker

Society for Computational Economics Working Group No. 1 on Language and Formal Semantics

August 25, 2026

Abstract

Economists increasingly use large language models (AIs) to write, modify, and translate modeling code, and AI output is probabilistic by construction. Unverified, such code can compute a model different from the one intended; the same prompts put to a different model, or to a later version of the same one, can yield a substantially different implementation, with no independent standard against which to judge either. Other computational sciences met this problem before AI made it acute, and each answered it the same way: a written statement of the model's meaning, separate from any implementation. Systems biology has SBML and its Systems Biology Ontology, physiology CellML, neuroscience NeuroML, engineering Modelica, and ecology the ODD protocol for agent-based models. We call such a statement a 'semantic ontology': an explicit, machine-readable account, independent of code and solver, of which economic objects exist, what stands for what, and what is written down, together with the conditions under which that interpretation holds. We propose a special issue collecting semantic ontologies for modeling languages, toolkits, empirical methods, and model classes.

1 The Challenge: AI and the Meaning of Economic Models

Economic theory is exact about its own objects. What is left implicit is the map from that theory to the things economists actually compute with: when a model is solved, its *meaning* is distributed across prose, notation, calibration, code, and tradition. The paper says "we solve the following model," but its equations underdetermine both what its code computes and what the computations mean.[1] Only an informed reader can fill the gaps, one who knows the surrounding literature and has absorbed the conventions and traditions the paper is built upon. The same gaps confront anyone who attempts to interpret the model's findings, rebuild the model, or extend it. As a result, without a concrete statement of the model's meaning to check against, computational results are hard to cross-verify, and models are hard to move between implementations.

Relying on implicit professional convention is costly even among humans: conventions are absorbed rather than stated, so anyone outside the tradition that produced a model must reconstruct them or guess. It fails outright when an AI writes, modifies, or translates modeling code – not because a language model knows no tradition, but because it has absorbed every tradition at once (even when they may conflict). Where an economist may know which subfield's conventions govern the model in front of them, an AI interpolates across conventions that conflict, settles the ambiguity silently, and returns fluent code bearing no trace that a choice was made; asked again, it may settle the ambiguity differently. To interpret or verify AI output, we need a statement, independent of the code, of which economic objects are computed and which relations among them are enforced; we will call such statements *semantic ontologies*. Outside economics such statements are established practice: systems biology, engineering, and neuroscience each maintain one (Hucka et al. 2003; Modelica Association 2023; Gleeson et al. 2010; see the *Semantic Ontologies* section of the annex). With semantic ontologies, a language model can translate a model between toolkits, and even draft new modeling research, with the output checked cheaply against the stated meaning; without them, there's no guarantee that the AI understood the model properly or produced what the user intended.

2 The Proposed Special Issue

We propose a special issue that demonstrates the construction and use of semantic ontologies for the domains of computational economics: domain-specific modeling languages (for example, Dynare), toolkits (for example, HARK and SSJ), empirical methods (for example, structural modeling and estimation), and model classes (for example, life-cycle, real-

[1] A made-up example, from no particular model or paper: consider that the sentence "a policy shock raises entrants at date t" is satisfied by two different models, because the shock may reach the cross-sectional distribution at t or at $t + 1$. The ambiguity becomes a problem when the code computes one of the two timing conventions and nothing in the paper records which. Both halves of the underdetermination are documented. Su and Judd (2012) recast one estimator in two computational formulations and recover identical estimates, so the two programs compute the same object; a semantic ontology makes that equivalence a stated, checkable claim. Dubé, Fox and Su (2012) find that a loose inner-loop tolerance moves estimated own-price elasticities by roughly a factor of two, so the code has quietly stopped computing the estimator the paper defines; a stated ontology, by naming the object computed and the conditions under which an approximation counts as computing it, makes such a divergence detectable rather than silent.

business-cycle, and agent-based models). In addition to the semantic ontologies, the working group will write a paper and assemble a community resource of canonical models and their ontologies.[2]

The working group will also contribute papers that use the stated ontologies to compare modeling strategies for the same problem — for instance, agent-based against heterogeneous-agent models.

Economics comes to this task with an unusual advantage — the theory already states the objects and relations an ontology must record, so what remains is the map from model files and toolkit calls to the theory — and an unusual obstacle: no settled timing convention to standardize against. Each team will therefore state the conventions its own domain assumes, and where two teams' ontologies disagree they will have located a point where the profession's conventions silently diverge — the substance of the closing comparison paper.

The SCE working group will submit a subset of the papers, an open call will invite the rest, and a comparison paper, written jointly by the participating teams, will close the special issue.

This format is familiar at the journal, whose scope includes computational methods: in two earlier special issues, several teams computed the same models and a closing comparison drew the results together (den Haan, Judd and Juillard 2010, 2011). We keep the many teams and the closing comparison, but the teams state what their models and code mean rather than solve a model in common. Climate modeling supplies the precedent for the comparison itself: the CF conventions fix what each quantity in model output means (Hassell et al. 2017), and that shared standard is what makes the recurring CMIP model intercomparisons possible.

The issue produces research infrastructure: once a model written for one toolkit can be read, checked, and re-solved in another, the ontologies outlast the papers. The semantic ontologies will be kept in an open-source GitHub repository, to which the community can contribute.

3 Organization

Before a paper is submitted, it is read within the working group by the authors of two other papers, against a checklist drawn from the research themes in the annex. The object is coherence across the issue rather than quality control: a referee assesses a paper on its own and has no way to see whether the ontologies use the same terms in the same senses, or state their conditions at comparable levels of rigour, which is what the closing comparison paper depends on. These reviews are advisory. They reach the guest editors as context, and

[2]The nearest precedent in economics, the Macroeconomic Model Data Base (Wieland et al. 2012), compares models under common variables, common shocks, and a menu of common policy rules while each model keeps its own equations. However the database does not standardize the semantic meaning of the models.

the working group can decline to submit a paper but cannot accept one. Papers then pass through the journal's ordinary external refereeing, which decides. We expect six to eight papers, with a session at the Society's conference between submission and revision.

3.1 Teams and Editors

The working group's members include developers of Dynare (Sébastien Villemot), HARK (Chris Carroll and Akshay Shanker), the VFI Toolkit (Robert Kirkby), GDSGE (Dan Cao) and QuantEcon (Matt McKay). Committed teams: [to be listed; only teams that have agreed in writing]. The working group's papers will come from its subgroups. The guest editors (a lead editor and at least one co-editor, neither submitting to the issue) are drawn from outside the working group. Their names go to the journal separately rather than in this document, which circulates: a prospective guest editor should hear of a nomination from the journal, not from a draft. Papers are submitted through the journal's editorial system, marked for this special issue, and are refereed under the journal's ordinary standards; the guest editors handle every paper, including the comparison paper, subject to the journal's final editorial authority.

3.2 Timeline

Calendar dates TBC on acceptance.

- Call for papers [on acceptance]
- Presentation at the next Computing in Economics and Finance meeting
  - This will be a working draft
  - There will be at least two special sessions devoted to the papers
- Submission draft incorporating CEF feedback within 2 months
- Internal review completed [Submission+3 months]
- Referee reports [Review+4 months]
- Revisions and the comparison paper [+24 months]

Annex: Guidelines for Papers

Semantic Ontologies

The goal of the semantic ontologies we propose is to codify formally the meaning of computational models. For our purposes, a semantic ontology has three components: what is assumed to exist in the domain of study (the *ontology*), what stands for what (the *denotation*), and what is written down (the *syntax*: a file, a model write-up, a specification).[3] The three components are exactly what any language definition must contain — a semantic domain, a semantic mapping, and a syntax (Harel and Rumpe 2004) — and the formalisms for stating them, surveyed under Methodology, come from both the ontology tradition and the mathematical semantics of programming languages. Theory supplies part of the semantic ontology (a general-equilibrium model's objects are precisely defined), but the semantic ontology must collect these definitions, map them concretely to computational and written counterparts, and state the assumptions under which the mapping holds. Without a written semantic ontology, nothing says which of the theory's objects a given file or function call stands for, and ordinary solver code gives one executable realization, not a solver-independent statement of what the representation denotes.

Other computational sciences already state their models' meaning separately from their computational implementations. Systems biology comes closest to what this issue proposes: SBML fixes the written form of biochemical-network models (Hucka et al. 2003), the Systems Biology Ontology states what the components of such a model mean (Courtot et al. 2011), and the community has maintained the two separately for over twenty years so that meaning can evolve without breaking syntax. Physiology curates CellML models with biophysical annotation in the Physiome repository (Yu et al. 2011); neuroscience's NeuroML gives data-driven neuron and network models a common description (Gleeson et al. 2010); Modelica attaches declarative equation semantics to physical-system models (Fritzson and Engelson 1998; Modelica Association 2023); and ecology's ODD protocol standardizes the description of agent-based models (Grimm et al. 2006). Manufacturing's Process Specification Language (Grüninger and Menzel 2003) and planning's PDDL2.1 (Fox and Long 2003) answer the same need for processes and plans. Compared with all of these, economics starts from an unusual advantage and an unusual obstacle. The advantage is that the relations among a domain of study's objects are the theory itself, and economics states its theory explicitly; what is missing is only the map from the representations economists compute with (model files, toolkit calls, estimation specifications) to that theory, and the conditions under which the map holds. The obstacle is that every field above could standardize against a fixed point its discipline had already settled — Modelica's component equations answer to conservation laws no physicist disputes — and economics has

[3] In a heterogeneous-agent model, for instance, the ontology contains the response of decision rules to prices, the cross-sectional distribution decisions induce, and the feedback of that distribution into prices, both as relations of the theory and as their computational counterparts.

no comparable fixed point. Competing equilibrium concepts, unsettled timing conventions, and toolkits built on incompatible primitives are the ordinary condition of the field rather than defects to be tidied away, and an ontology that presumed them away would describe no one's practice. The consequence is not that the exercise is hopeless but that it changes character: the ontologies must be *stated* rather than discovered, each team committing to the conventions its own domain assumes. Where two teams' ontologies then disagree, they have located a point at which the profession's conventions silently diverge — a result of the issue rather than a failure of it, and the substance of the closing comparison. Stating these maps, domain by domain, is the work this issue proposes.

Research Domains and Themes

Domains of study. Most papers in the special issue will address one *domain of study*. Domains of study can include a modeling language, a toolkit, an empirical method, a model class, or a combination of these. Some will be illustrations of the use of the semantic ontologies to compare existing models to each other. We anticipate that there may be a few other closely-related papers; for example, one working group participant is eager to write a guide for economists for how to use the 'lean' programming language for development of both code and theory.[4]

DOMAINS OF STUDY

- **Modeling languages.** A modeling language is a fixed grammar in which a complete model is written as a file, as in Dynare's model language or Dolo's YAML model files.
- **Toolkits.** A toolkit is a collection of construction calls and classes from which a model is assembled in a programming language, as in HARK's agent classes, the sequence-space Jacobian toolkit (Auclert et al. 2021), or the VFI Toolkit.
- **Estimation and empirical methods.** An empirical method takes a model to data. It states which measured objects stand for the model's quantities (for example, prices and shocks), which classifications organize those objects, and which transformations construct the inputs to estimation or calibration. The simulated method of moments and indirect inference are examples of empirical methods, and so is the calibration of a computable general-equilibrium model to a social accounting matrix or input-output table.

[4] Whichever domain a paper addresses, its theory and mathematics are taken as pre-existing: the Bellman equation is not in question, while what a given Dynare file or HARK model means is. A language, toolkit, or empirical-method paper therefore emphasizes the map from representations to the mathematical objects they stand for, while a model-class paper emphasizes the objects and relations implied by its mathematical and theoretical framework. Differences among the stated ontologies are results for the closing comparison to report.

- **Model classes.** A model class is a family of models built from the same kinds of objects, as in heterogeneous-agent macroeconomies or overlapping-generations economies.

Research themes. The papers developing a semantic ontology will establish the ontology's *metatheory*. There are three components to each semantic ontology: what is assumed to exist (the *ontology*), what stands for and relates to what (the *denotation*), and what is written down (the *syntax*). Domains of study will differ in how important these components are, and some, such as a model class, may not have certain components, such as syntax. The metatheory states when the interpretation holds (*well-posedness*) and what preserves it (*equivalence*, *adequacy*, and *convergence*). Details of these research themes are given in the Research Themes section at the end of this annex.

Methodology

Formalizing the semantic ontology. A requirement for each semantic ontology paper will be that it presents a *formal* account of the ontology of its domain of study. However, the researchers choose the formalism in which they state these components.

The available formalisms differ in how much meaning they fix, and they can be ordered from least to most (Uschold and Grüninger 1996).

At one end, a team records which entities exist and which relations connect them, as boxes and arrows: an entity-relationship diagram (Chen 1976), a UML class diagram (Berardi, Calvanese and De Giacomo 2005), or a knowledge graph (Hogan et al. 2021). These *graphs and diagrams* name the model's objects but place few restrictions on what they mean.

Logical axioms fix more: they rule out interpretations of the objects much as parameter restrictions rule out models. The axioms can be stated in any formal logic. In the older tradition this meant first-order logic (Gruber 1993), and in much current practice it means a description logic such as the Semantic Web's Web Ontology Language (OWL) (Baader et al. 2017; W3C OWL Working Group 2012).

The *mathematical semantics* developed for programming languages fixes the most meaning: it attaches an explicit mathematical object to every written form. Denotational semantics assigns each form an object in the ordered structures of domain theory (Scott and Strachey 1971). Initial-algebra semantics takes meaning to be the unique homomorphism from a many-sorted algebra of terms (Goguen, Thatcher, Wagner and Wright 1977). Typed categories interpret a typed syntax in a category with matching structure (Lambek and Scott 1986). Operational semantics gives rules for execution on an abstract machine (Plotkin 1981), and axiomatic semantics gives the assertions that hold before and after execution (Hoare 1969).

Practice differs by field. Google and other technology firms record knowledge graphs from observed data, without axioms (Noy et al. 2019). The Gene Ontology Consortium states its ontology of gene functions in description logic (Ashburner et al. 2000). A few programming languages, such as Standard ML and WebAssembly, have been given complete formal semantics (Milner, Tofte, Harper and MacQueen 1997; Haas et al. 2017), though most languages in use have not.

How far along this scale a paper goes depends on whether its domain of study has a syntax. A team whose domain has one (a modeling language or a toolkit) can use any of the mathematical semantics; a team whose ontology has no syntax (a model class, or an empirical method before its specification is constructed) can axiomatize the ontology in a logic or record it as a graph. Ecology's ODD protocol, a description standard for agent-based models maintained across two decades, shows that a domain without a syntax can still be described precisely enough that others can rebuild its models (Grimm et al. 2020). Declarative languages make the mathematical route easiest: a Dynare or Dolo file states the model itself rather than a procedure for solving it, so the file is already the kind of written form a semantic map can interpret. A team seeking machine-checked guarantees can go further and formalize its domain in a proof assistant such as Lean, defining the model objects as types and the file-to-object map as a function inside the system, so that well-posedness and equivalence become theorems the machine verifies (de Moura and Ullrich 2021).

Top-down vs. bottom-up approaches. A team can also build its semantic ontology top-down or bottom-up. Building top-down, it starts from the theory and formalizes the theory's objects; building bottom-up, it starts from observed data and relations and abstracts the ontology from them, as industry does when it reconstructs a conceptual model from a data catalog (AWS Database Blog 2026). In this issue we expect economic theory to supply the structure, so most papers will work top-down; a bottom-up construction may suit agent-based modeling or policy research, where observed relations precede a settled theory.

Research Themes

A paper addresses those of the following five research themes relevant to its domain of study. The first three are the components of the domain's semantic ontology; the last two are statements about it (when the interpretation holds, and what preserves it under changes of representation and computation):

The research themes

1. **Ontology: what is assumed to exist.** The economic and mathematical entities and relations within the domain (agents, states, shocks, timing, the equilibrium concept, operators, functions, and which objects determine which), together with the criteria for when two of them are the same (Gruber 1993; Guarino, Oberle and Staab 2009). The ontology is stated in its own terms, without reference to the syntax.

2. **Denotation: what stands for what.** The meaning of the ontology's objects. The meaning takes the form of relationships between the mathematical, economic and computational representations, for instance value functions as economic concepts, instantiated on a mathematical space, and approximated on the computer. Typically the denotations will also include relationships between these representations and the written forms: each written form stands for its mathematical or economic object under an explicit map (Harel and Rumpe 2004).

3. **Syntax: what is written down.** The set of legal written forms: the model file, the sequence of construction calls, or the specification of an estimation exercise. For instance, a language has a full grammar; a toolkit may expose construction calls or classes. A domain (such as an empirical method) may have no syntactic written form at all, in which case much of the role of syntax is played by the denotation.

The next two research themes are not components of the semantic ontology; they are statements about it:

4. **Well-posedness: when the interpretation holds.** The conditions under which the denotation is well defined: domains, units, timing, information structure, and the parameter restrictions assumed.

5. **Equivalence, adequacy, convergence: what preserves it.** Denotational equivalence: two written forms, one denotation. Adequacy: an implementation computes exactly the object denoted. Convergence: a numerical approximation approaches it as grids and tolerances are refined.

References

Ashburner, M., C. A. Ball, J. A. Blake, et al. (2000). "Gene Ontology: tool for the unification of biology." *Nature Genetics* 25(1), 25–29. [link]

Auclert, A., B. Bardóczy, M. Rognlie, and L. Straub (2021). "Using the Sequence-Space Jacobian to Solve and Estimate Heterogeneous-Agent Models." *Econometrica* 89(5), 2375–2408. [link]

AWS Database Blog (2026). "Build a semantic ontology to power AI assistants on AWS — Part 1." 14 July 2026. Accessed 29 July 2026. [link]

Baader, F., I. Horrocks, C. Lutz, and U. Sattler (2017). *An Introduction to Description Logic*. Cambridge University Press. [link]

Berardi, D., D. Calvanese, and G. De Giacomo (2005). "Reasoning on UML Class Diagrams." *Artificial Intelligence* 168(1–2), 70–118. [link]

Chen, P. P. (1976). "The Entity-Relationship Model — Toward a Unified View of Data." *ACM Transactions on Database Systems* 1(1), 9–36. [link]

Courtot, M., N. Juty, C. Knüpfer, et al. (2011). "Controlled vocabularies and semantics in systems biology." *Molecular Systems Biology* 7, 543. [link]

de Moura, L., and S. Ullrich (2021). "The Lean 4 Theorem Prover and Programming Language." In *Automated Deduction — CADE 28*, Lecture Notes in Artificial Intelligence 12699. Springer, 625–635. [link]

den Haan, W. J., K. L. Judd, and M. Juillard (2010). "Computational suite of models with heterogeneous agents: Incomplete markets and aggregate uncertainty." *Journal of Economic Dynamics and Control* 34(1), 1–3. [link]

den Haan, W. J., K. L. Judd, and M. Juillard (2011). "Computational suite of models with heterogeneous agents II: Multi-country real business cycle models." *Journal of Economic Dynamics and Control* 35(2), 175–177. [link]

Dubé, J.-P., J. T. Fox, and C.-L. Su (2012). "Improving the Numerical Performance of Static and Dynamic Aggregate Discrete Choice Random Coefficients Demand Estimation." *Econometrica* 80(5), 2231–2267. [link]

Fox, M., and D. Long (2003). "PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains." *Journal of Artificial Intelligence Research* 20, 61–124. [link]

Fritzson, P., and V. Engelson (1998). "Modelica — A Unified Object-Oriented Language for System Modeling and Simulation." In *ECOOP '98 — Object-Oriented Programming*, Lecture Notes in Computer Science 1445. Springer, 67–90. [link]

Gleeson, P., S. Crook, R. C. Cannon, M. L. Hines, G. O. Billings, et al. (2010). "NeuroML: A Language for Describing Data Driven Models of Neurons and Networks with a High Degree of Biological Detail." *PLoS Computational Biology* 6(6), e1000815. [link]

Goguen, J. A., J. W. Thatcher, E. G. Wagner, and J. B. Wright (1977). "Initial Algebra Semantics and Continuous Algebras." *Journal of the ACM* 24(1), 68–95. [link]

Grimm, V., U. Berger, F. Bastiansen, et al. (2006). "A standard protocol for describing individual-based and agent-based models." *Ecological Modelling* 198(1–2), 115–126. [link]

Grimm, V., S. F. Railsback, C. E. Vincenot, et al. (2020). "The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism." *Journal of Artificial Societies and Social Simulation* 23(2), 7. [link]

Gruber, T. R. (1993). "A translation approach to portable ontology specifications." *Knowledge Acquisition* 5(2), 199–220. [link]

Guarino, N., D. Oberle, and S. Staab (2009). "What Is an Ontology?" In S. Staab and R. Studer (eds.), *Handbook on Ontologies*, 2nd ed. Springer, 1–17. [link]

Grüninger, M., and C. Menzel (2003). "The Process Specification Language (PSL): Theory and Applications." *AI Magazine* 24(3), 63–74. [link]

Haas, A., A. Rossberg, D. L. Schuff, B. L. Titzer, D. Gohman, L. Wagner, A. Zakai, J. F. Bastien, and M. Holman (2017). "Bringing the Web up to Speed with WebAssembly." In *PLDI 2017*. ACM, 185–200. [link]

Harel, D., and B. Rumpe (2004). "Meaningful modeling: what's the semantics of 'semantics'?" *Computer* 37(10), 64–72. [link]

Hassell, D., J. Gregory, J. Blower, B. N. Lawrence, and K. E. Taylor (2017). "A data model of the Climate and Forecast metadata conventions (CF-1.6) with a software implementation (cf-python v2.1)." *Geoscientific Model Development* 10(12), 4619–4646. [link]

Hoare, C. A. R. (1969). "An axiomatic basis for computer programming." *Communications of the ACM* 12(10), 576–580. [link]

Hogan, A., E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, et al. (2021). "Knowledge Graphs." *ACM Computing Surveys* 54(4), article 71. [link]

Hucka, M., A. Finney, H. M. Sauro, et al. (2003). "The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models." *Bioinformatics* 19(4), 524–531. [link]

Lambek, J., and P. J. Scott (1986). *Introduction to Higher Order Categorical Logic*. Cambridge University Press. [link]

Milner, R., M. Tofte, R. Harper, and D. MacQueen (1997). *The Definition of Standard ML (Revised)*. MIT Press.

Modelica Association (2023). *Modelica Language Specification*, version 3.6. [link]

Noy, N., Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor (2019). "Industry-scale knowledge graphs: lessons and challenges." *Communications of the ACM* 62(8), 36–43. [link]

Plotkin, G. D. (1981). *A Structural Approach to Operational Semantics*. Report DAIMI FN-19, Computer Science Department, Aarhus University.

Scott, D., and C. Strachey (1971). "Toward a mathematical semantics for computer languages." *Proceedings of the Symposium on Computers and Automata*, Polytechnic Institute of Brooklyn, 19–46. [link]

Su, C.-L., and K. L. Judd (2012). "Constrained Optimization Approaches to Estimation of Structural Models." *Econometrica* 80(5), 2213–2230. [link]

Uschold, M., and M. Grüninger (1996). "Ontologies: principles, methods and applications." *The Knowledge Engineering Review* 11(2), 93–136. [link]

W3C OWL Working Group (2012). *OWL 2 Web Ontology Language Document Overview*, 2nd ed. W3C Recommendation. [link]

Wieland, V., T. Cwik, G. J. Müller, S. Schmidt, and M. Wolters (2012). "A new comparative approach to macroeconomic modeling and policy analysis." *Journal of Economic Behavior & Organization* 83(3), 523–541. [link]

Yu, T., C. M. Lloyd, D. P. Nickerson, et al. (2011). "The Physiome Model Repository 2." *Bioinformatics* 27(5), 743–744. [link]