SO: A Foundational Ontology for Computational Sociology

Dr Edit Hlaszny
Dr Hlaszny Bioystems Engineering
Mail: edit@edithlaszny.eu
http://www.edithlaszny.eu/

Abstract

The history of computational sociology extends over the last 4.5 decades; its roots can perhaps be found in general systems theory and structural functionalism. Ontologies have been created in a wide range of subject areas and their number and application areas are dramatically growing. However, it can be considered quite well-founded to assume that no ontology has been created in the general sociological subject area so far. The SO (sociological ontology) mentioned in the title makes a modest attempt at this, hoping that true experts in the subject area will find the topic itself (creating and further developing sociological ontologies) interesting. Therefore, let us quote modestly the esteemed Basel mathematician Johann Bernoulli, Opera Omnia, 67, Tom. I.: *"Problema novum ad cuius solutionem sociologi invitantur."*

Keywords

Computational sociology; ontology; Java-based application.

1. What's Actually in This Paper

1.1 For readers . . .

who'd rather cut to the chase than plough through the usual academic throat-clearing, here's what has actually been built:

- *The ontology itself:* 700+ classes covering general sociology, 254 object properties, 78 data properties, and 201 n-ary causal relations—all properly mapped under BFO 2020 top-level ontology classes.
- *A web-based browser* that lets you navigate the whole thing without getting lost in the conceptual weeds.
- *Development tools* for extending the system: Eclipse IDE integration, 60 Java classes, database management scripts, and a LaTeX-based documentation generator–because ontologies that can't evolve are rather pointless.

The paper walks through the theoretical foundations (skip if you're in a hurry), technical implementation (don't skip if you're planning to use this), and demonstrates why computational sociology might benefit from having its conceptual house in order.

Crack on—read what interests you, skip what doesn't :)

2. On computational Sociology

2.1 The Epistemological Foundations of Sociological Ontology

The fundamental methodological divide between social and natural sciences stems from their divergent subject matter and analytical challenges. Natural sciences examine phenomena governed by universal laws, enabling prediction and replication through controlled experimentation.

Social sciences confront human agency, cultural variation, and historical contingency, rendering absolute prediction impossible. Social phenomena emerge from complex interactions between individual choice and structural constraints, creating inherently interpretive challenges.

The observer-observed relationship further complicates social inquiry. Researchers cannot achieve complete detachment from their cultural context, whilst their subjects possess reflexive awareness that can alter behaviour under study.

Mathematical formalisation has historically correlated with scientific maturity across disciplines. Physics achieved predictive precision through mathematical modelling, whilst chemistry and biology developed rigorous quantitative frameworks as they matured.

However, this relationship requires nuanced evaluation. Mathematics provides analytical precision and enables hypothesis testing, yet its applicability varies across domains. Economics extensively employs mathematical methods whilst remaining contentious regarding predictive accuracy.

The presumption that mathematical sophistication equals scientific validity risks privileging quantification over explanatory depth. Complex social phenomena may resist meaningful reduction to mathematical representations without losing essential characteristics.

An elaborated sociological ontology would represent a significant advancement in computational sociology by providing systematic conceptual architecture for social phenomena. Traditional computational approaches often suffered from ad hoc categorisations and inconsistent terminology.

A rigorous ontological framework enables precise definition of social concepts, their relationships, and hierarchical organisation. This facilitates automated reasoning, knowledge integration, and comparative analysis across diverse sociological domains.

Ontological standardisation promises enhanced reproducibility in computational social research. Researchers can build upon shared conceptual foundations rather than constructing idiosyncratic frameworks for each investigation.

The SO system demonstrates how formal ontological methods can capture sociological complexity whilst maintaining logical consistency. By grounding social concepts within established philosophical frameworks like BFO, it bridges humanistic insight with computational tractability.

Such developments suggest computational sociology's evolution from purely quantitative analysis towards sophisticated conceptual modelling. This represents methodological advancement rather than replacement of traditional sociological approaches.

The integration of ontological reasoning with empirical analysis may ultimately transcend the quantitative-qualitative divide by providing structured frameworks for both numerical data and interpretive understanding within unified analytical systems.

2.2 The Crucial Role of Ontologies in the Modern Era

In an age defined by the exponential growth of information, the discipline of ontology has transcended its philosophical origins to become a cornerstone of modern knowledge engineering. Ontologies, as formal specifications of a shared conceptualization, serve as foundational frameworks for structuring, organizing, and interpreting information in a machine-readable manner.

They provide a precise and unambiguous vocabulary of classes, properties, and relationships to model a domain of interest. This semantic rigor is essential or transforming unstructured data into meaningful, interconnected knowledge graphs. Beyond simple data categorization, ontologies enable advanced forms of reasoning and inference, allowing computational systems to discover new relationships, validate logical consistency, and make informed decisions that would be impossible with traditional data models. The role of ontologies today is therefore not merely descriptive but is actively generative, creating the semantic infrastructure necessary for intelligent systems to operate effectively in an increasingly complex world.

3. The Semantic Web and Its Foundational Technologies

3.1 Technological overview

The vision of the Semantic Web, as an extension of the World Wide Web, is to make Internet data machine-readable and semantically meaningful, facilitating seamless integration and automated reasoning. Its technical foundation is built upon a layered architecture of standards and languages designed to achieve this goal. At the core are resource description frameworks such as RDF (Resource Description Framework), which provides a simple, graph-based model for making statements about resources in the form of subject-predicate-object triples.

For expressing more complex relationships and formal axioms, OWL (Web Ontology Language) serves as the primary language. OWL offers a rich set of constructors for defining classes, properties, and the intricate logical relationships between them, enabling sophisticated reasoning over the data.

The Web Ontology Language has three sublanguages (OWL Lite, OWL DL, and OWL Full) each offering different levels of expressiveness and corresponding reasoning capabilities. These ontologies are often encoded in standardized formats such as OWL/XML, RDF/XML, or JSON-LD, ensuring interoperability across different tools and platforms.

The ecosystem of semantic technologies includes reasoners (such as FaCT++ and HermiT) that perform logical inference and consistency checks, query languages like SPARQL that enable complex graph queries, and various API libraries and software frameworks that facilitate the development and manipulation of semantic data. Collectively, these technologies provide a robust and powerful toolkit for building and leveraging semantic representations of knowledge.

3.2 The Social Ontology (SO) in Practice

The Social Ontology (SO) represents a formal and systematic conceptualization of the domain of sociology. It provides a foundational vocabulary of classes and properties for modeling the full spectrum of social phenomena, from micro-level interactions and individual dispositions to macro-level social structures, institutions, and global processes. By rigorously defining these concepts and their relationships, the SO serves as a powerful instrument for:

Research:
It enables researchers to formalize hypotheses and theories in a machine-readable format, facilitating automated reasoning and the discovery of non-obvious connections between disparate social concepts. For instance, a researcher could use the ontology to query for all bfo:processes that occur in a bfo:realizable entity (like Law), or to analyze how different forms of Social_Control (realizable entity) are linked to various types of Deviance (process). This level of formalization supports quantitative, qualitative, and mixed-methods research by providing a common semantic ground.

Education:
As a pedagogical tool, the SO can be used to teach students the foundational concepts and theoretical frameworks of sociology in a structured and interconnected way. It visually represents the relationships between different schools of thought, key concepts, and their hierarchical organization, providing a clear and comprehensive map of the discipline.

Interoperability:
The SO's alignment with a robust upper ontology like BFO 2020 ensures that its concepts can be semantically integrated and reasoned over with other domain ontologies in fields such as public health, economics, political science, and environmental science. This allows for a trans-disciplinary understanding of complex societal issues, where sociological insights can be linked with data and knowledge from other fields to create a more holistic and powerful knowledge base for research, policy-making, and social analysis.

4. Basic design considerations of SO: object properties

4.1 Object properties and their inverses

The existence of an object property does not logically entail the existence of its inverse - this would require an additional axiom or inference rule. However, the situation with reasoners is more subtle than it might initially appear.

The Reasoner Perspective
Some reasoners (particularly those implementing complete tableaux algorithms for description logics like ALCIQ or SHOIN) can indeed work with implicit inverse relationships without requiring explicit inverse property declarations. They achieve this through:

Query rewriting: When encountering a pattern that would benefit from an inverse property, they can reformulate queries to use the original property in reverse

Internal inverse handling: They maintain internal representations that treat $\mathtt{P}(\mathtt{x},\mathtt{y})$ and $\mathtt{P}^{-1}(\mathtt{y},\mathtt{x})$ as equivalent without requiring explicit declaration of $\mathtt{P}^{-1}$. However, this capability varies significantly across reasoners and reasoning tasks.

4.2 The Case for Explicit Inverse Properties

Despite some reasoners' capabilities, there are compelling reasons to define inverse properties explicitly:

Semantic Clarity:
Inverse properties often represent genuinely distinct conceptual relationships. In a sociology ontology, "hasChild" and "hasParent" aren't merely logical inverses - they capture different social and conceptual perspectives on kinship relations.

Reasoner Agnosticism:
Not all reasoners handle implicit inverses equally well, particularly when dealing with complex property chains, transitivity, or functional properties.

Query Expressiveness:
Explicit inverses enable more natural and efficient SPARQL queries and API interactions.

Ontological Completeness:
If a relationship is conceptually bidirectional and both directions are meaningful in your domain, explicit representation better captures the ontological structure.

A Pragmatic Approach:
Many people believe that some kind of pragmatic approach is worth taking: Rather than defining inverses for "every possible case," they use this heuristic. *Define explicit inverses when:* The inverse represents a conceptually distinct relationship (hasChild/hasParent), when the inverse is frequently queried, or when it participates in different axioms or property characteristics. *Rely on implicit inverses when:* The inverse is purely a logical convenience with no distinct conceptual content.

4.3 Computational overhead

For SO, relationships like "belongsToOrganization / hasMember" or "influencedBy / influences" likely warrant explicit inverse properties due to their conceptual significance in sociological analysis.

The computational overhead of additional properties is generally negligible compared to the benefits in semantic expressiveness and reasoning reliability.

Nevertheless, the SO includes inverse properties for each object property, ensuring that future ontologists developing the system have these inverses readily available and can choose whether to utilise them or not. This hierarchy provides a comprehensive foundation for sociological object properties. Each object property includes both the forward and inverse relationships, along with concise annotations explaining their sociological significance.

4.4 Object properties follow sociological dimensions

- *Communication:* the mechanism through which social reality is constructed and maintained
- *Cultural relations:* shared meaning systems that bind communities
- *Demographic properties:* essential for population and life course classes

- *Deviance and control properties:* crucial for extensive deviance and crime classes
- *Economic relations:* material basis of many social relationships and inequalities
- *Environmental properties:* important your spatial and urbanization classes
- *Health/medical properties:* important for medical sociology concepts
- *Membership and belonging:* essential for group dynamics and identity formation
- *Organizational properties:* important for many institutional and organizational classes
- *Political properties:* necessary for governance and power-related classes
- *Power and influence:* fundamental to understanding social stratification and control
- *Research methodology properties:* essential for connecting research-related classes (Census, Interview, Observation, etc.)
- *Social control:* mechanisms for maintaining order and transmitting culture
- *Social relationships:* the building blocks of social networks and community
- *Spatial/temporal relations:* contextual factors shaping social interaction
- *Stratification properties:* vital for social class and inequality concepts
- *Technology/media properties:* increasingly crucial for contemporary sociology

These entities (the SO contains a total of 254 object properties) work well with the class hierarchy (674 classes) of the ontology, particularly with entities like Social_Organizations, Social_Groups, Social_Processes, and the various institutional categories.

4.5 Object Property Characteristics

The ontology browser displays all the characteristic bits of the object properties in a table. Characteristics can be

Functional

A functional object property is a relationship that, for a given subject, can have only one object. In other words, if an entity $\mathcal{A}$ is related to an entity $\mathcal{B}$ via a functional property $\mathcal{P}$, then $\mathcal{A}$ cannot also be related to any other entity $\mathcal{C}$ via that same property $\mathcal{P}$ (unless $\mathcal{C}$ is the same as $\mathcal{B}$). This enforces a one-to-one or many-to-one relationship from subject to object.

Inverse Functional

An inverse functional object property is a relationship where the inverse of the property is functional. This means that if an entity $\mathcal{A}$ is related to an entity $\mathcal{B}$ via a property $\mathcal{P}$, then $\mathcal{B}$ cannot also be the object of that same property $\mathcal{P}$ for any other entity $\mathcal{C}$ (unless $\mathcal{C}$ is the same as $\mathcal{A}$. This enforces a one-to-one or one-to-many relationship from subject to object.

Transitive

A transitive object property is a relationship where if an entity $\mathcal{A}$ is related to $\mathcal{B}$ via the property $\mathcal{P}$, and $\mathcal{B}$ is related to $\mathcal{C}$ via $\mathcal{P}$, then the reasoner can automatically infer that $\mathcal{A}$ is also related to $\mathcal{C}$ via $\mathcal{P}$.

Symmetric

A symmetric object property is a relationship where if an entity $\mathcal{A}$ is related to $\mathcal{B}$ via the property $\mathcal{P}$, then the reasoner automatically infers that $\mathcal{B}$ is also related to $\mathcal{A}$ via $\mathcal{P}$.

Asymmetric

An asymmetric object property is a relationship where if an entity $\mathcal{A}$ is related to $\mathcal{B}$ via the property $\mathcal{P}$, it is automatically inferred that $\mathcal{B}$ cannot be related to $\mathcal{A}$ via $\mathcal{P}$. This is a stronger condition than simply being irreflexive, as it states that the inverse relationship is impossible.

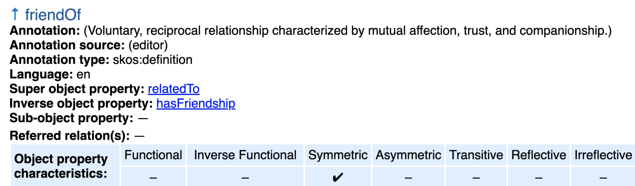
Reflexive

A reflexive object property is a relationship where every individual is related to itself via that property. This is a less common characteristic in sociological modeling.

Irreflexive

no individual can be related to itself via that property. This is a very common characteristic for many relationships in sociology.

This interpretation of object properties is also intended to help shed more light on the ontology browser data.


↑ friendOf
Annotation: (Voluntary, reciprocal relationship characterized by mutual affection, trust, and companionship.)
Annotation source: (editor)
Annotation type: skos:definition
Language: en
Super object property: relatedTo
Inverse object property: hasFriendship
Sub-object property: —
Referred relation(s): —

Object property characteristics:	Functional	Inverse Functional	Symmetric	Asymmetric	Transitive	Reflective	Irreflective
	–	–	✔	–	–	–	–

Figure 1: Object property characteristics showing in the browser.

5. The N-ary Relation Framework

5.1 A Formal Architecture for Sociological Causation

The representation of complex sociological causation within ontological structures presents fundamental challenges to standard binary relationship models. The framework implemented addresses this through a rigorous reification pattern that preserves both the collective nature of causal mechanisms and the multiplicity of their consequences whilst maintaining computational tractability.

Sociological phenomena characteristically exhibit causal relationships wherein multiple factors operate conjointly to produce various effects. A binary predicate structure, limited to expressing relations between precisely two entities, proves inadequate for capturing such complexity. The n-ary relation framework transcends this limitation through systematic reification—the explicit representation of relationships themselves as first-class ontological entities.

5.2 Architectural Components

The framework comprises five interrelated structural elements:

1. Reified Causal Events: Each n-ary relationship instantiates a unique individual of the class Collective_Causal_Event. This reified entity serves as the ontological anchor for the entire causal structure, providing a singular referent for what is conceptually a complex, multi-participant relationship.
2. Contributory Linkages: Domain elements—those entities functioning as causal antecedents—connect to the reified event through the object property contributesToCausalEvent. This establishes their role as conjoint causal factors whilst maintaining their individual identities within the larger causal mechanism.
3. Consequential Linkages: Range elements—the effects or outcomes—connect from the reified event via the object property causalEventProduces. This directional relationship preserves the causal asymmetry inherent in sociological explanation whilst accommodating multiple simultaneous consequences.
4. Cartesian Product Assertions: The framework generates exhaustive binary assertions between each domain element and each range element using the primary relational predicate. This provides direct access paths for reasoning engines whilst preserving semantic transparency for domain experts unfamiliar with reification patterns.
5. Semantic Annotations: Each reified event individual carries structured annotations documenting the theoretical basis, empirical support, and sociological significance of the causal relationship, transforming the formal structure into interpretable scholarly knowledge.

5.3 Epistemological Advantages

This architecture offers several methodological benefits for computational sociology:

1. The reification pattern enables attachment of meta-level properties—certainty measures, evidential support, temporal scope—to relationships themselves rather than merely to participating entities. This captures the epistemological status of causal claims with appropriate granularity.
2. The dual representation—both reified events and direct Cartesian assertions—accommodates varying degrees of ontological sophistication amongst users. Domain experts may query direct relationships intuitively whilst formal reasoners exploit the richer reified structure.
3. The framework maintains extensibility: additional causal factors or consequences may be incorporated without restructuring existing relationships, supporting the iterative refinement characteristic of sociological theory development.

5.4 Formal Semantics

Let $D = \{d_1, d_2, \ldots, d_n\}$ represent domain entities and $R = \{d_1, d_2, \ldots, d_m\}$ represent domain entities represent range entities in a causal relationship mediated by predicate P. The framework generates:

1. A unique reified individual (Figure 2):
$e \in Collective\_Causal\_Event$
2. Contributory assertions: (Figure 3):
$contributesToCausalEvent(d_i, e)$ for all $d_i \in D$
3. Consequential assertions: (Figure 4):
$causalEventProduces(e, r_j)$ for all $r_j \in R$
4. Direct assertions: (Figure 5):
$P(d_i, r_j)$ for all $d_i \in D$ and $r_j \in R$

This generates n contributory linkages, m consequential linkages, and $n \times m$ direct assertions, providing multiple inference paths whilst maintaining logical coherence.

5.5 An OWL/XML encoded example

```
<!--
    n-ary causal relation:
        Youth Mobilization and Social Change

    {Youth_Culture, Social_Movement} → mobilizes →
    {Social_Change, Culture}
  -->

<!--  freshing reified individual   -->
<Declaration>
    <NamedIndividual IRI="#causal_Event_ID000201"/>
</Declaration>
<ClassAssertion>
    <Class IRI="#Collective_Causal_Event"/>
    <NamedIndividual IRI="#causal_Event_ID000201"/>
</ClassAssertion>

<AnnotationAssertion>
    <AnnotationProperty abbreviatedIRI="skos:definition"/>
     <IRI>#causal_Event_ID000201</IRI>
    <Literal>The causal_Event_ID000201 named individual is
             associated to the relation: 'Youth Mobilization
             and Social Change'. The individual represents
             a reified causal event capturing the sociological
             defined by the name of the relation and the
             symbolic description. The causal_Event_ID000201
             is named individual of the Collective_Causal_Event
             class, with identifiers reflecting their systematic
             role in modeling complex n-ary sociological
             relationships.
    </Literal>
</AnnotationAssertion>
```

Figure 2: n-ary causal relation: "Youth Mobilization and Social Change". Generating a unique reified individual for each n-ary causal relation. Each individual is annotated.

```
<!--  connecting causes to event  -->
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#contributesToCausalEvent"/>
    <NamedIndividual IRI="#youth_Culture_ID000002"/>
    <NamedIndividual IRI="#causal_Event_ID000201"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#contributesToCausalEvent"/>
    <NamedIndividual IRI="#social_Movement_ID000014"/>
    <NamedIndividual IRI="#causal_Event_ID000201"/>
</ObjectPropertyAssertion>
```

Figure 3: Connecting causes to event.

```
<!--  connecting event to effects  -->
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#causalEventProduces"/>
    <NamedIndividual IRI="#causal_Event_ID000201"/>
    <NamedIndividual IRI="#social_Change_ID000016"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#causalEventProduces"/>
    <NamedIndividual IRI="#causal_Event_ID000201"/>
    <NamedIndividual IRI="#culture_ID000009"/>
</ObjectPropertyAssertion>
```

Figure 4: Connecting event to effects.

```
<!--  direct Cartesian product assertions  -->
<ObjectPropertyAssertion>
    <ObjectProperty  IRI="#mobilizes"/>
    <NamedIndividual IRI="#youth_Culture_ID000002"/>
    <NamedIndividual IRI="#social_Change_ID000016"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty  IRI="#mobilizes"/>
    <NamedIndividual IRI="#youth_Culture_ID000002"/>
    <NamedIndividual IRI="#culture_ID000009"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty  IRI="#mobilizes"/>
    <NamedIndividual IRI="#social_Movement_ID000014"/>
    <NamedIndividual IRI="#social_Change_ID000016"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty  IRI="#mobilizes"/>
    <NamedIndividual IRI="#social_Movement_ID000014"/>
    <NamedIndividual IRI="#culture_ID000009"/>
</ObjectPropertyAssertion>
```

Figure 5: Direct Cartesian product assertions.

5.6 Implementation Significance

The successful deployment of this framework across 201 causal relationships within the Subject Ontology demonstrates its practical viability. The architecture proves computationally tractable—reasoning completes within acceptable timeframes—whilst providing the semantic richness required for sophisticated sociological analysis.

This reification pattern represents a principled solution to the representation of complex causation in formal ontologies, balancing theoretical rigour with practical utility. It provides computational sociology with infrastructure capable of expressing the nuanced, multi-factorial causal relationships that characterise social phenomena, thereby advancing the discipline's capacity for formal knowledge representation and automated reasoning.

After all, this constitutes the crux of ontological work where genuine domain expertise proves essential. The technical framework provides the infrastructure, but its true value emerges only when populated with relationships grounded in rigorous sociological theory and empirical research.

Whilst the author has endeavoured to implement a sound computational architecture, the substantive sociological content—the selection of causal factors, the theoretical justification of relationships, the nuanced understanding of social mechanisms—requires depth of disciplinary knowledge that can only come from established scholars in the field.

It is here, in the realm of sociological interpretation rather than technical implementation, that the author's limitations become most apparent, and where collaboration with domain experts would prove invaluable.

6. Technical Details

6.1 A Formal Architecture for Sociological Causation