Formal Structures in Systems Architecture

Towards Ontology-Informed Mathematical Foundations

Monte Mahlum

monte-mahlum.github.io

January 7, 2026

Section 1

Introduction

Introduction

Motivation

Definition

A *system architecture* is a conceptual model defining the structure, behavior, and views of a given system[1]. It is the language in which requirements and specifications are declared and verified.

Hannu Jaakkola and Bernhard Thalheim. (2011) "Architecture-driven modelling methodologies." In: Proceedings of the 2011 conference on Information Modelling and Knowledge Bases XXII. Anneli Heimbürger et al. (eds). IOS Press. p. 98

.

Introduction

Motivation

Definition

A *system architecture* is a conceptual model defining the structure, behavior, and views of a given system[1]. It is the language in which requirements and specifications are declared and verified.

Hannu Jaakkola and Bernhard Thalheim. (2011) "Architecture-driven modelling methodologies." In: Proceedings of the 2011 conference on Information Modelling and Knowledge Bases XXII. Anneli Heimbürger et al. (eds). IOS Press. p. 98

.

Motivation

- Complex systems (or systems-of-systems) are ubiquitous, and undergoing dramatic transformation.
- Evolution of complex systems is slow, difficult, and often ad hoc.
- An abundance of data and mathematical tools can enable novel architectural reasoning for societal good.

Introduction

Programmatic Perspective

Methodology

1. Develop a rich ontology for describing systems.
2. Define categorical structures grounded in this ontology.
3. Deploy mathematical tools to ...

Introduction

Programmatic Overview

Methodology

1. Develop a rich ontology for describing systems.
2. Define categorical structures grounded in this ontology.
3. Deploy mathematical tools to ...

Envisioned Utility

- validate system requirements from primitive data,
- detect, avoid, and analyze anomalous or adverse behavior,
- predict novel capabilities,
- rapidly evolve complex systems in a *safe*, *responsible*, and *interpretable* manner.

Brief Literature Review

Previous Work – Joint between NASA LaRC and JHU APL.

1 M.M., Samantha Jarvis, Nelson Niu, Angeline Aguinaldo, Amanda Hicks, and Ian Levitt. *Formal Structures in Systems Ontology towards Air Traffic Management Architectures*. NASA Technical Memorandum, 2025. Report no. NASA/TM-20250010771.

Sample of Related Work

2 Patrick Schultz and David I. Spivak. *Temporal Type Theory: A Topos-Theoretic Approach to Systems and Behavior*. Birkhäuser, 2019. DOI 10.1007/978-3-030-00704-1.

3 Patrick Schultz, David I. Spivak, and Christina Vasilakopoulou. *Dynamical Systems and Sheaves*. arXiv preprint, 2016. arXiv:1609.08086. DOI 10.48550/arXiv.1609.08086.

4 Gioele Zardini, David I. Spivak, Andrea Censi, and Emilio Frazzoli. *A Compositional Sheaf-Theoretic Framework for Event-Based Systems*. arXiv preprint, 2021. arXiv:2101.10485. DOI 10.48550/arXiv.2101.10485.

5 Sophie Libkind and David Jaz Myers. *Towards a double operadic theory of systems*. arXiv preprint, 2025. arXiv:2505.18329. DOI 10.48550/arXiv.2505.18329.

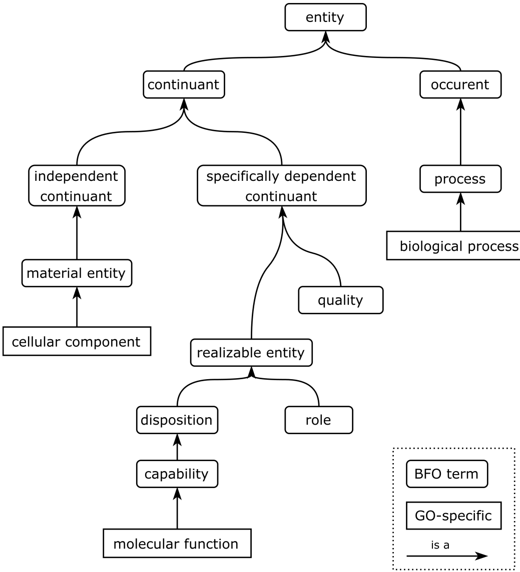
Section 2

Ontological Foundations

Ontological Foundations

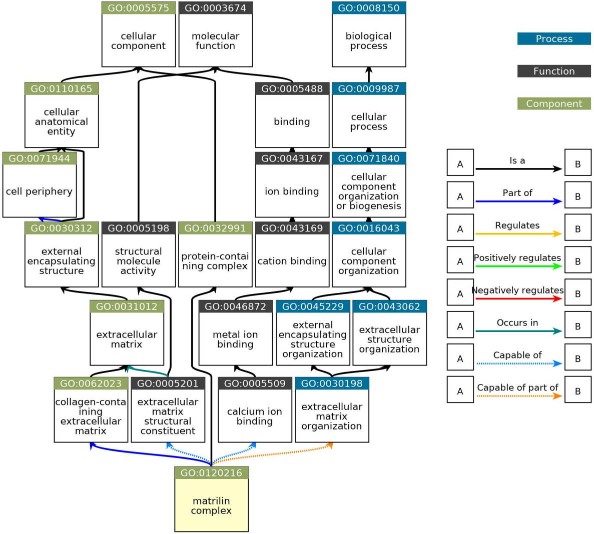
The Basic Formal Ontology (BFO)

Figure: A sample of terms from BFO. The vocabulary is further connected by relations not shown here, e.g., every process has participating independent continuants.



Ontological Foundations

The Gene Ontology (GO) as Inspiration



QuickGO - https://www.ebi.ac.uk/QuickGO

Figure: A sample of terms from the Gene Ontology and their relation to *matrilin complex* (bottom), a type of cellular component.

Sourced from EMBL's European Bioinformatics Institute, CC0, via Wikimedia Commons

Ontological Foundations

The Systems Ontology (SO)

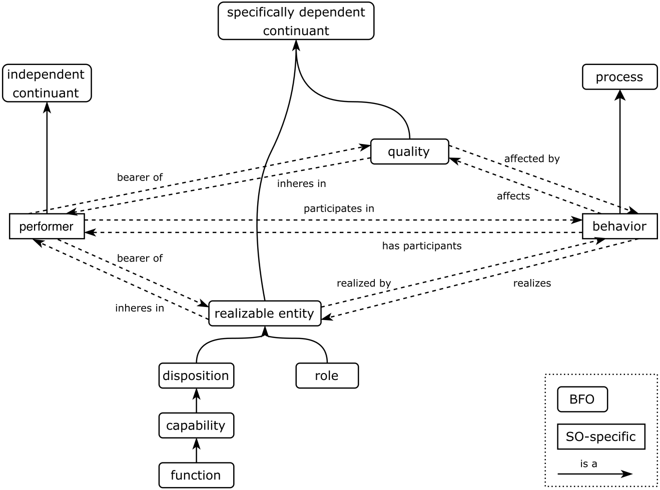


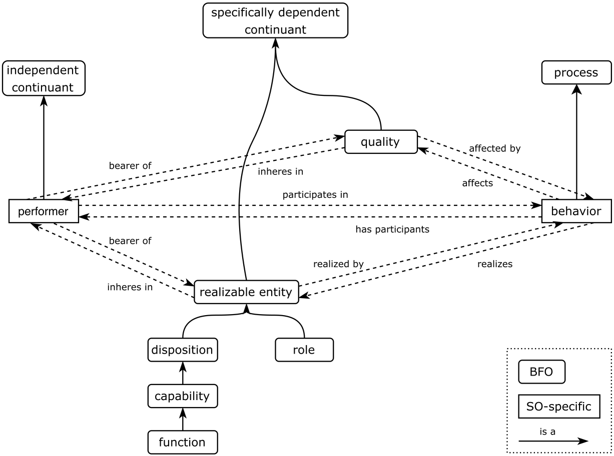
Figure: The core terms and relations of SO, including the central realization pattern (lower dashed triangle).

Ontological Foundations

The Systems Ontology (SO)

Interpretation

Systems Ontology is a lightweight *framework* more than a complete ontology in itself.



Ontological Foundations

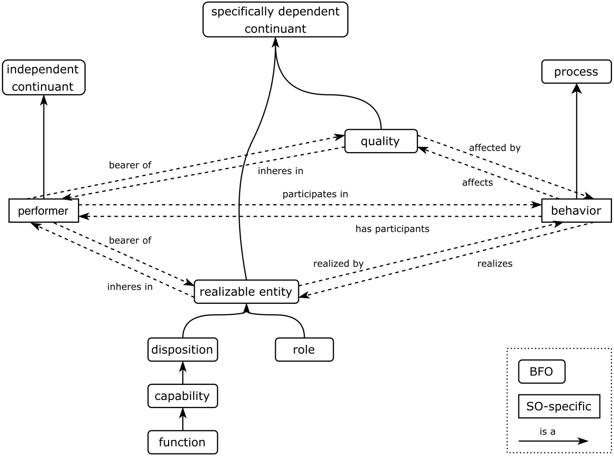
The Systems Ontology (SO)

Interpretation

Systems Ontology is a lightweight *framework* more than a complete ontology in itself.

Intended Utility

A double-categorical schema to build domain-specific system ontologies.

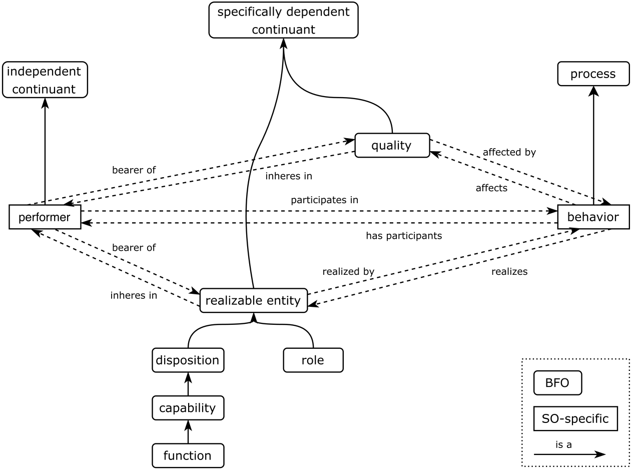


Ontological Foundations

The Systems Ontology (SO)

Examples of Possible Performer Types

- pilot or plane,
- computer,
- measurement tool,
- cellular component (GO),
- any other class of physical components deemed relevant to the system,

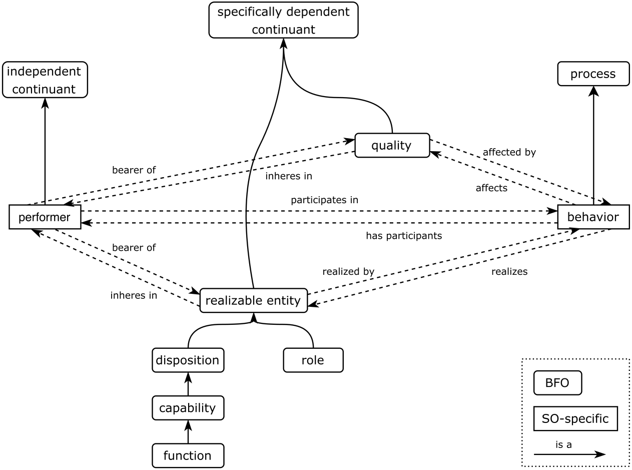


Ontological Foundations

The Systems Ontology (SO)

Examples of Possible Behavior Types

- take off,
- high-altitude maneuver,
- photovoltaic conversion,
- signal transduction (GO),
- DNA replication (GO)



Ontological Foundations

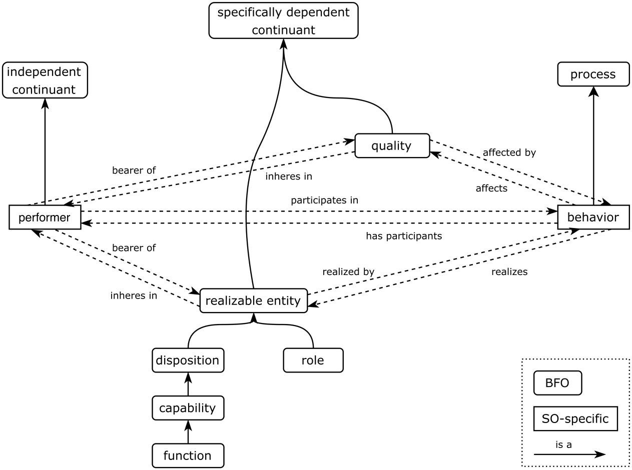
The Systems Ontology (SO)

Examples of Possible (Universal) Qualities

- temperature,
- voltage,
- protein configuration,
- location and velocity (although not standard BFO qualities).

Definition

A *quality space* S_Q for a quality Q is a set of possible values that a measurement of Q can take, e.g., if $Q = \mathrm{temperature}$, S_Q may be $\mathbb{R}_{\geq 0}$ interpreted in units of Kelvin.

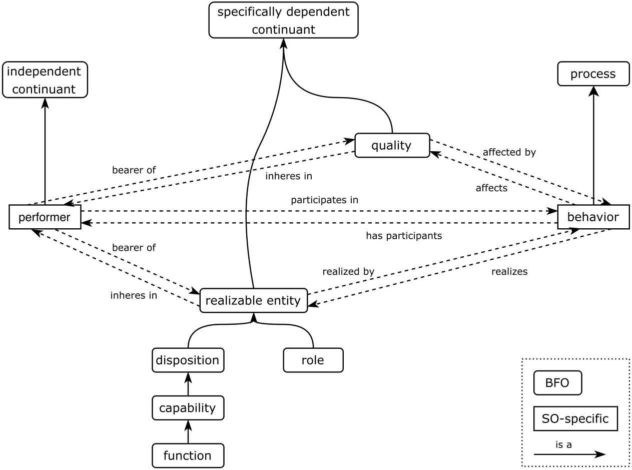


Ontological Foundations

The Systems Ontology (SO)

Examples of Possible Capability Types

- emergency water landing,
- *safe* emergency water landing,
- photovoltaic conversion,
- photovoltaic conversion with efficiency $\geq e$.
- signaling receptor binding (GO).



Section 3

Ontology-Informed Mathematical Structures

Ontology-Informed Mathematical Structures

The Performer Hierarchy

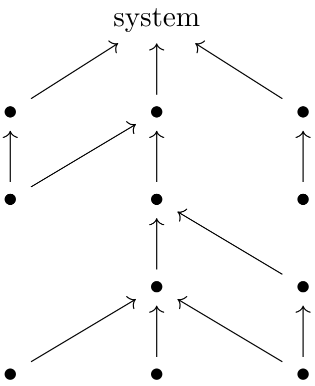
Definition

Let $\mathbf{P}$ denote the poset (or lattice, or frame), whose objects are those performers in the system with a morphism $p \to q$ if and only if p is *physically contained in* q.

All posets are treated as sites with the *minimal coverage* where $(p_i \to p)_i$ covers p if and only if for all minimal $q \to p$, $q \to p_i$ for some i. Here q is minimal if $q' \to q$ implies $q' = q$ or $\emptyset$.

Remark

While every $p \in \mathbf{P}$ will have a declared type $P \xrightarrow{\text{is a}} \boxed{\text{performer}}$, the structure $\mathbf{P}$ is distinct from $\downarrow \boxed{\text{performer}}$.



Ontology-Informed Mathematical Structures

Quality Sheaves and their Morphisms

Definition

Let $\mathscr{Q} : \mathbf{P}^{\mathrm{op}} \to \mathbf{Set}$ be a sheaf declaring quality spaces[1], and thus qualities, for each performer.

Remark

$\mathscr{Q}$ defines a state space $\mathscr{Q}(\mathrm{system})$ and records how these states descend to local regions of the system.

$$
\begin{tikzcd}
& \prod_{i\in I_1} S_{Q_i} \arrow[dl] \arrow[d] \arrow[dr] & \\
\bullet \arrow[d] & \bullet \arrow[dl] \arrow[d] & \bullet \arrow[d] \\
\bullet & \prod_{i\in I_2\cup I_3} S_{Q_i} \arrow[d] \arrow[dr] & \bullet \\
& \prod_{i\in I_2} S_{Q_i} \arrow[dl] \arrow[d] \arrow[dr] & \prod_{i\in I_3} S_{Q_i} \arrow[d] \\
\bullet & \bullet & \bullet
\end{tikzcd}
$$

[1] For each (universal) quality $Q \xrightarrow{\text{is a}} \boxed{\text{quality}}$, there are many possible "quality spaces" S_Q which are sets of possible measurement values for Q each with respect to a different unit of measurement.

Ontology-Informed Mathematical Structures

Quality Sheaves and their Morphisms

Definition

Let $\mathscr{Q} : \mathbf{P}^{\mathrm{op}} \to \mathbf{Set}$ be a sheaf declaring quality spaces[1], and thus qualities, for each performer.

Remark

$\mathscr{Q}$ defines a state space $\mathscr{Q}(\text{system})$ and records how these states descend to local regions of the system.

[1]For each (universal) quality $Q \xrightarrow{\text{is a}} \boxed{\text{quality}}$, there are many possible "quality spaces" S_Q which are sets of possible measurement values for Q each with respect to a different unit of measurement.

Covariant Grothendieck Construction

$(\mathbf{P}, \mathscr{Q}) \in \int^{\mathrm{cov}}_{\mathbf{FinPos}^{\mathrm{op}}} \mathrm{Sh}(-)$ where

$$\begin{aligned} \mathrm{Sh}(-) : \mathbf{FinPos}^{\mathrm{op}} &\to \mathbf{Cat} \\ \mathbf{P} &\mapsto \mathrm{Sh}(\mathbf{P}) \\ (\mathbf{P} \xrightarrow{f} \mathbf{P}') &\mapsto f_* \end{aligned}$$

A *quality sheaf morphism*

$$(\rho, \alpha) : (\mathbf{P}, \mathscr{Q}) \to (\mathbf{P}', \mathscr{Q}')$$

is then a pair

$$\begin{aligned} \rho : \mathbf{P}' \to \mathbf{P} \quad &\text{in } \mathbf{FinPos} \\ \alpha : \rho_* \mathscr{Q} \to \mathscr{Q}' \quad &\text{in } \mathrm{Sh}(\mathbf{P}') \end{aligned}$$

Ontology-Informed Mathematical Structures

Quality Sheaves and their Morphisms

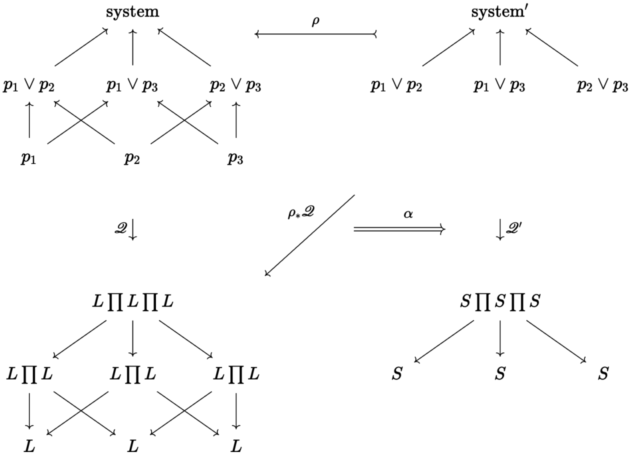


Figure: Depiction of a quality sheaf morphism enabling abstraction. One interpretation of this morphism is as follows: the performers p_i denote airplanes in an airspace system, the quality space L is for the *aircraft location* quality, while $S = \{\text{unsafe}, \text{ safe}\}$ is for the pairwise *seperation safety* quality.

Ontology-Informed Mathematical Structures

Behaviors & Capabilities

Guiding Principle

While behaviors and capabilities are ontologically distinct, they should be represented in the same mathematical language as a transformation of a quality sheaf.

Behaviors & Capabilities

Integrating Diverse Representations

Guiding Principle

While behaviors and capabilities are ontologically distinct, they should be represented in the same mathematical language as a transformation of a quality sheaf.

Difficulty

There are many disparate means of modeling processes:

- natural language specification (assumptions and guarantees),
- black box relation,
- wiring diagram,
- dynamical system, etc.

Ontology-Informed Mathematical Structures

Behaviors & Capabilities

Guiding Principle

While behaviors and capabilities are ontologically distinct, they should be represented in the same mathematical language as a transformation of a quality sheaf.

Difficulty

There are many disparate means of modeling processes:

- natural language specification (assumptions and guarantees),
- black box relation,
- wiring diagram,
- dynamical system, etc.

Primary Issue

Capabilities often expressed in natural language using an abstract quality sheaf $\mathscr{Q}'$ while raw system data and behavioral reasoning (e.g., nontrivial concurrency) often requires a highly specific and time-dependent representation, $\mathscr{Q}(\text{system})^{\mathbb{R}}$.

Ontology-Informed Mathematical Structures

Behaviors & Capabilities

Guiding Principle

While behaviors and capabilities are ontologically distinct, they should be represented in the same mathematical language as a transformation of a quality sheaf.

Difficulty

There are many disparate means of modeling processes:

- natural language specification (assumptions and guarantees),
- black box relation,
- wiring diagram,
- dynamical system, etc.

Primary Issue

Capabilities often expressed in natural language using an abstract quality sheaf $\mathscr{Q}'$ while raw system data and behavioral reasoning (e.g., nontrivial concurrency) often requires a highly specific and time-dependent representation, $\mathscr{Q}(\mathrm{system})^{\mathbb{R}}$.

Future Work

Integrate the quality sheaf framework with behavioral theories such as Temporal Type Theory [Schultz, Spivak] or DOTS [Libkind, Jaz Myers] to enable passage from low-level behaviors to the high-level capabilities that they realize.

References

1 M.M., Samantha Jarvis, Nelson Niu, Angeline Aguinaldo, Amanda Hicks, and Ian Levitt. *Formal Structures in Systems Ontology towards Air Traffic Management Architectures*. NASA Technical Memorandum, 2025. Report no. NASA/TM-20250010771. https://ntrs.nasa.gov/citations/20250010771

2 Robert Arp, Barry Smith, and Andrew D. Spear. *Building Ontologies with Basic Formal Ontology*. The MIT Press, 2015. ISBN 9780262527811. https://mitpress.mit.edu/9780262527811/building-ontologies-with-basic-formal-ontology/

3 Evan Patterson. *Knowledge Representation in Bicategories of Relations*. arXiv preprint, 2017. arXiv:1706.00526. DOI 10.48550/arXiv.1706.00526. https://arxiv.org/abs/1706.00526

4 Alexandru E. Stanculescu. *On calculus of functors in model categories*. arXiv preprint, 2012. arXiv:1208.1919. DOI 10.48550/arXiv.1208.1919. https://arxiv.org/abs/1208.1919

5 Patrick Schultz and David I. Spivak. *Temporal Type Theory: A Topos-Theoretic Approach to Systems and Behavior*. Birkhäuser, 2019. DOI 10.1007/978-3-030-00704-1. https://link.springer.com/book/10.1007/978-3-030-00704-1

6 Patrick Schultz, David I. Spivak, and Christina Vasilakopoulou. *Dynamical Systems and Sheaves*. arXiv preprint, 2016. arXiv:1609.08086. DOI 10.48550/arXiv.1609.08086. https://arxiv.org/abs/1609.08086

7 Gioele Zardini, David I. Spivak, Andrea Censi, and Emilio Frazzoli. *A Compositional Sheaf-Theoretic Framework for Event-Based Systems*. arXiv preprint, 2021. arXiv:2101.10485. DOI 10.48550/arXiv.2101.10485. https://arxiv.org/abs/2101.10485

8 Sophie Libkind and David Jaz Myers. *Towards a double operadic theory of systems*. arXiv preprint, 2025. arXiv:2505.18329. DOI 10.48550/arXiv.2505.18329. https://arxiv.org/abs/2505.18329