Architecture of Complex Systems

Alexey A. Nekludoff
AstraVerge Research

E-mail: an@astraverge.org
ORCID: 0009-0002-7724-5762

17 January 2026

Abstract

This paper introduces the *Architecture of Complex Systems (ACS)* as an ontological framework for understanding systems as coherent architectures rather than as collections of behaviors or dynamic processes. ACS addresses a foundational question: under what conditions does a system exist as a unified entity at all?

The framework is grounded in the primacy of relations over observation and dynamics. Architectural truth is defined as relational and is shown to depend on coherence, closure, and invariants within a bounded relational structure. On this basis, the paper establishes a fundamental boundary: no amount of observational data or knowledge of system dynamics suffices to reconstruct system ontology. Dynamics may describe state transitions and trajectories, but they do not uniquely determine architectural structure.

A minimal relational algebra is introduced as a working language for architectural reasoning. This algebra is used to articulate admissibility, coherence, and architectural existence without reference to domain-specific implementations or empirical realizations. Architecture is treated as ontologically prior to both dynamics and observation.

The contribution of this work is intentionally foundational. It fixes the ontological core of ACS and situates it within a broader research program concerned with discrete existence and the limits of observation. Domain-specific interpretations and applications are left as subsequent extensions built upon this explicitly defined foundation.

Keywords: architecture of complex systems; architectural ontology; relational structure; architectural coherence; system existence; admissibility; relational algebra; architecture and dynamics; ontological non-reconstructibility; limits of observation.

Contents

1 Introduction **1**

2 Architecture and Ontology **2**

3 Ontological Core of the Architecture of Complex Systems **2**
3.1 Axiom 1: Relational Primacy 2
3.2 Axiom 2: Coherence Condition 3
3.3 Axiom 3: Observational Insufficiency 3
3.4 Axiom 4: Ontological Non-Reconstructibility 3
3.5 Axiom 5: Non-Uniqueness of Ontological Mapping 4
3.6 Summary 4

4 Relational Algebra as a Working Language of Architecture **4**
4.1 Relations as Architectural Constraints 4
4.2 Composition of Relations 5
4.3 Closure and Architectural Boundaries 5
4.4 Coherence of Relational Structures 5
4.5 Invariants and Architectural Integrity 5
4.6 Admissibility and the Space of Possibility 5
4.7 Architecture and Algebraic Reasoning 5
4.8 Relational Operators 6
4.9 Theorems and Consequences 6

5 Architecture and Dynamics **7**
5.1 States, Transitions, and Architectural Space 8
5.2 Reversibility and Structural Underdetermination 8
5.3 Dynamics as Selection Within Admissibility 8
5.4 Functional Equivalence and Ontological Distinction 8
5.5 Architectural Degradation and Dynamic Continuity 8
5.6 Ontological Priority of Architecture 9
5.7 Architectural Plurality and Non-Uniqueness 9

6 Relation to Foundational Frameworks **9**
6.1 Architecture and Discrete Being 9
6.2 Architecture and the Limits of Observation 10
6.3 Structural Continuity Across Frameworks 10

7 Conclusion and Outlook **10**

A Notation and Conventions **12**

1 Introduction

The study of complex systems has traditionally focused on their observable behavior and dynamic evolution. Across disciplines, systems are commonly characterized through trajectories, time series, event streams, or other forms of measurable dynamics. While such approaches have proven effective for prediction and control, they leave a fundamental question insufficiently addressed: under what conditions does a system exist as a coherent entity at all?

In many contemporary frameworks, architectural structure is treated as secondary to behavior. System identity is inferred from input–output relations, execution traces, or statistical regularities. This observational stance implicitly assumes that sufficient data can, in principle, restore the underlying organization of a system. However, this assumption fails once architectural coherence is lost or was never explicitly defined.

This work starts from a different premise. We argue that the existence of a system is determined not by its dynamics, but by the relational architecture that constrains and stabilizes its states. Dynamics describe how a system evolves; architecture determines whether a system exists as a unified structure. Observable behavior may be reversible, reproducible, or predictable, while the architectural conditions that make such behavior possible remain fundamentally underdetermined by observation alone.

The aim of this paper is to introduce the *Architecture of Complex Systems (ACS)* as an ontological framework. ACS defines the relational conditions under which complex systems exist, persist, and degrade as coherent architectures. Rather than proposing another descriptive model of system behavior, ACS specifies the structural prerequisites of architectural truth.

The contribution of this work is deliberately limited. We do not address domain-specific implementations, technological realizations, or empirical case studies. Instead, we establish the ontological core of ACS and clarify the distinction between architectural structure, system dynamics, and observational data. Domain-specific extensions are treated as subsequent developments rather than foundational components.

ACS does not aim to replace domain-specific theories, but to specify the architectural conditions under which such theories meaningfully apply.

Ontological Premise: Localities and Coherence

Complex systems do not exist as undifferentiated wholes. They exist as configurations of localities—bounded domains within which relations are defined, constraints apply, and internal consistency may be maintained. Locality is therefore not a spatial or technical notion, but an ontological condition of distinguishability. Without locality, neither relations nor architectural structure can be meaningfully defined.

The existence of a whole presupposes coherence between localities. Coherence denotes the absence of contradiction between the conditions that govern different local domains. Only when local constraints are mutually compatible can a composite structure exist as a unified entity. Incoherent configurations may exhibit activity or change, but they do not constitute a system in the architectural sense.

A detailed ontological analysis of locality and coherence is developed within the framework of the *Philosophy of Discrete Being* (FDB) [2]. The present work does not require acceptance of FDB as a complete ontological system. It adopts only a minimal premise: that architectural coherence presupposes locality. This premise establishes the ontological ground on which architectural reasoning becomes possible.

On this basis, the Architecture of Complex Systems (ACS) is developed as a constructive continuation. While FDB addresses the conditions under which localities exist and may be coherent, ACS specifies how coherent localities can be structurally connected to form complex systems. Coherence, at the level of ACS, is treated not as an ontological axiom but as a structural property of relations between localities. A system exists architecturally only insofar as these relations are composed without contradiction.

This separation of levels is deliberate. FDB provides the ontological conditions of existence. ACS provides the architectural conditions of constructibility. Together, they allow stable forms to be understood not merely as outcomes of dynamics, but as entities whose existence depends on explicitly definable architectural constraints.

2 Architecture and Ontology

The term *architecture* is often used ambiguously. In practice, it may refer to implementation choices, component layouts, or design documentation. In theoretical contexts, it is frequently conflated with system behavior or functional organization. ACS adopts a stricter interpretation: architecture denotes the relational structure that defines the conditions of system existence.

Ontology, in this context, concerns what it means for a system to exist as a coherent entity. An ontological description specifies the entities, relations, and constraints that constitute a system independently of how it is observed or how it evolves over time. Architecture, understood ontologically, is not a representation of a system but a condition of its being.

A crucial distinction must therefore be drawn between architecture and dynamics. Dynamics describe transitions between states. They may be continuous or discrete, deterministic or stochastic, reversible or irreversible. Architecture, by contrast, specifies which states and transitions are admissible in the first place. It defines the space within which dynamics can occur, rather than the trajectories that unfold within that space.

Observation introduces a further layer. Observational data capture manifestations of system behavior: measurements, events, traces, or outcomes. While such data may constrain hypotheses about system structure, they do not uniquely determine it. Multiple ontologically distinct architectures may give rise to indistinguishable dynamics under observation. Consequently, architectural truth cannot be reduced to observational adequacy.

ACS therefore treats architecture as ontologically prior to both dynamics and observation. A system is architecturally real only insofar as its relations form a coherent and closed structure. When relational coherence is violated or absent, no amount of observational data can restore architectural truth. This priority of architecture over dynamics and observation forms the basis for the ontological core developed in the following sections.

3 Ontological Core of the Architecture of Complex Systems

This section introduces the ontological core of the Architecture of Complex Systems (ACS). The core is formulated through a minimal set of axioms that define the conditions under which a system exists as a coherent architectural entity. These axioms are not empirical generalizations and do not depend on domain-specific interpretations. They specify structural conditions of existence rather than observable properties or dynamic behavior.

The axioms are introduced in a deliberate order. Each subsequent axiom refines or constrains the implications of the preceding ones. Together, they establish a closed ontological framework for architectural reasoning.

3.1 Axiom 1: Relational Primacy

Architectural truth is relational.

Architectural truth does not reside in isolated components, states, or observable outputs. It is determined by the relations that bind elements into a structured whole. Entities without relations possess no architectural significance; they do not contribute to the existence of a system as such. Architecture is therefore defined not by what elements are, but by how they are related.

This axiom establishes relations as the primary ontological constituents of architecture. Objects, components, or agents are architecturally meaningful only insofar as they participate in a relational structure. Without relations, there is no architecture to speak of.

3.2 Axiom 2: Coherence Condition

A system possesses architectural truth only if its relations compose into a coherent structure.

Not every collection of relations constitutes an architecture. For a system to exist architecturally, its relations must be composable without contradiction. Coherence is not introduced as an external logical requirement, but as an intrinsic property of relational composition. A relational structure is coherent if its composition admits a non-empty space of admissible configurations.

Coherence thus entails structural consistency, closure, and mutual compatibility of relations within a bounded whole. A partially specified or internally contradictory set of relations fails to define an architectural entity. Such configurations may exhibit activity or transient behavior, but they lack architectural existence.

Architectural existence is therefore treated as binary rather than gradual within the scope of ACS. A system either satisfies the coherence condition and exists as an architectural entity, or it fails to exist architecturally. Persistence over time, stability of behavior, or observational regularities do not compensate for the absence of coherent relational composition.

This axiom establishes coherence as a criterion of architectural existence. Architecture is not inferred from activity, persistence, or dynamics, but from the presence of a relational structure whose composition is free of contradiction.

3.3 Axiom 3: Observational Insufficiency

No amount of observational data can restore architectural truth once relational coherence is lost.

Observational data capture manifestations of system behavior: measurements, events, traces, or outcomes. While such data may constrain possible interpretations of a system, they do not constitute architecture itself. Observation operates at the level of appearances, not at the level of structural conditions.

Once relational coherence is absent or violated, architectural truth is destroyed. Because architecture is defined by relations rather than by their observable effects, no accumulation of data can reconstruct what is no longer structurally defined. Observational adequacy is therefore insufficient as a criterion of architectural existence.

This axiom establishes a strict boundary between architecture and observation. Architecture is ontologically prior to any observational account and cannot be recovered through observation alone.

3.4 Axiom 4: Ontological Non-Reconstructibility

The ontology of a system cannot be reconstructed from its dynamics.

Dynamics describe transitions between states. They characterize how a system changes over time, possibly in a deterministic, stochastic, reversible, or irreversible manner. However, dynamics presuppose the existence of a system; they do not define its ontological structure.

Even complete knowledge of system dynamics does not uniquely determine the relations that constitute the system. State trajectories may be invertible or predictable without revealing the architectural constraints that make such trajectories admissible. Structural organization is therefore not derivable from dynamic evolution.

This axiom formalizes the distinction between behavior and being. Dynamics concern how a system evolves; ontology concerns what a system is. The latter cannot be inferred from the former.

3.5 Axiom 5: Non-Uniqueness of Ontological Mapping

Dynamics do not uniquely determine ontology.

The mapping from system ontology to observable dynamics is not injective. Distinct relational architectures may give rise to indistinguishable or equivalent dynamic behavior. As a result, there exists no unique inverse mapping from dynamics to ontology.

This axiom expresses the formal consequence of the preceding ones. Architectural reconstruction from dynamics is not merely difficult or incomplete; it is fundamentally non-unique. Multiple ontologically distinct systems may be consistent with the same observed or inferred dynamics.

The non-uniqueness of ontological mapping establishes an intrinsic limit to reconstruction-based approaches. Architecture cannot be identified solely through functional equivalence or behavioral correspondence. It must be specified at the level of relational structure.

3.6 Summary

Together, these axioms define the ontological core of ACS. They establish relations as the locus of architectural truth, coherence as the condition of existence, and non-reconstructibility as a fundamental boundary. Architecture is thus prior to dynamics and observation, and systems exist architecturally only insofar as their relational structures are coherent and closed.

4 Relational Algebra as a Working Language of Architecture

This section introduces a minimal relational vocabulary employed in ACS as a working algebra for architectural reasoning. The purpose of this algebra is not computation, prediction, or description of behavior, but the formal expression of architectural constraints. It serves as a language for specifying the conditions of admissibility, coherence, and existence of systems as architectures.

The relational algebra used in ACS is intentionally minimal. It avoids domain-specific constructs and does not presuppose any particular physical, biological, social, or technical realization. Its role is ontological rather than operational: to articulate what must hold for a system to exist architecturally.

4.1 Relations as Architectural Constraints

In ACS, relations are treated as primary architectural entities. A relation is understood as a constraint that links entities within a system and restricts the space of admissible states. Relations do not describe interactions, events, or observations. They define conditions under which entities may coexist as parts of a coherent whole.

Architectural relations are not temporal. They do not specify sequences or causal chains. Instead, they establish structural dependencies that delimit what configurations of a system are possible or impossible. Architecture is therefore constituted by relations rather than by components taken in isolation.

4.2 Composition of Relations

Relations do not exist independently. Architectural structure arises through the composition of relations into higher-order configurations. Composition denotes the combination of relations such that the resulting structure imposes constraints not reducible to those of individual relations.

Relational composition is not sequential and does not imply temporal ordering. It expresses structural coupling: how constraints interact and jointly restrict admissible configurations. Through composition, architecture acquires depth and internal structure without reference to dynamic execution.

4.3 Closure and Architectural Boundaries

A central architectural notion is closure. A relational structure is closed if all relations required for its coherence are contained within the system. Closure defines the boundary of architectural existence.

If coherence depends on relations external to the defined structure, the architecture is ontologically incomplete. Such configurations may participate in processes or interactions, but they do not constitute a system as a closed architectural entity. Closure therefore distinguishes systems from open-ended processes and from aggregates lacking architectural unity.

4.4 Coherence of Relational Structures

Coherence refers to the mutual consistency of relations within a closed structure. A coherent architecture contains no relational contradictions and admits at least one non-empty set of admissible configurations.

Coherence is an ontological property rather than a dynamic one. It does not depend on persistence, stability, or longevity over time. A system may be dynamically unstable yet architecturally coherent, or dynamically persistent while lacking architectural coherence. Only the latter case corresponds to architectural non-existence.

4.5 Invariants and Architectural Integrity

Architectural invariants are relational conditions that must hold across all admissible states of a system. They express what architecture preserves regardless of dynamic evolution.

Invariants are not values or measurements. They are structural constraints that define architectural integrity. Violation of an invariant does not merely alter system behavior; it destroys architectural coherence. In this sense, invariants function as ontological guards rather than operational checks.

4.6 Admissibility and the Space of Possibility

Admissibility defines the set of states and transitions permitted by a given relational architecture. Architecture determines what is possible; dynamics select what is realized within that space.

The admissible space is defined entirely by relations, their composition, closure, coherence, and invariants. Dynamics operate within this space but do not define it. A change in admissibility corresponds to an architectural change, not merely to a dynamic transition.

4.7 Architecture and Algebraic Reasoning

The relational algebra introduced here provides a formal language for architectural reasoning. It allows architectural questions to be posed independently of observation and dynamics: questions of existence, coherence, and admissibility precede questions of behavior.

By employing a minimal algebra of relations, ACS maintains ontological generality while preserving formal rigor. Architecture is thus expressed as a structured space of constraints, within which complex systems may exist, evolve, or degrade.

4.8 Relational Operators

The relational algebra employed in ACS makes use of a minimal set of operators. These operators do not describe computation, execution, or temporal evolution. They express architectural constraints and conditions of admissibility that determine whether a system exists as a coherent architecture.

Composition. The primary operator of the relational algebra is composition. If R_1 and R_2 are architectural relations, their composition, denoted $R_1 \circ R_2$, represents the joint constraint resulting from their simultaneous applicability. Composition does not imply sequence, causation, or temporal order. It expresses the structural conjunction of constraints within an architectural context.

Composition is the operator through which architectural structures are formed. Coherence, contradiction, and admissibility are defined with respect to composed relations rather than isolated ones.

Restriction. Relations in ACS are always defined relative to localities. The restriction operator, denoted $R \upharpoonright L$, specifies the applicability of a relation R within a locality L. Restriction localizes architectural constraints and prevents relations from being treated as global abstractions.

Through restriction, architectural analysis can distinguish between local and global coherence, as well as between localized contradictions and system-wide architectural failure.

Compatibility. Compatibility expresses whether relations can be jointly satisfied. Two relations R_1 and R_2 are said to be compatible if their composition admits at least one admissible configuration. Architectural incompatibility indicates that no configuration can satisfy the composed constraints.

Compatibility is not a logical predicate but an architectural condition. It concerns the existence of admissible configurations rather than the truth of propositions. Coherence at the architectural level is characterized by the mutual compatibility of all composed relations.

Invariant Preservation. Architectural invariants express conditions that must hold across all admissible configurations of a system. The invariant preservation relation, denoted $R \models I$, indicates that a relational structure R preserves an invariant I across its admissible space.

Violation of an architectural invariant does not merely alter system behavior. It constitutes architectural degradation, in which the system loses its architectural identity while potentially retaining dynamic activity.

Together, these operators provide a minimal algebraic language for architectural reasoning. They allow architectural existence, coherence, and degradation to be expressed without reference to implementation, execution, or domain-specific realizations.

4.9 Theorems and Consequences

The following theorems formalize key consequences of the axioms of ACS using the relational operators introduced above. They do not introduce additional assumptions. Each theorem follows directly from the ontological core and the relational algebra of architectural constraints.

Theorem 1 (Non-Reconstructibility).

> *Given only system dynamics, the relational architecture of a system is not uniquely reconstructible.*

Proof sketch. Architectural structure is defined by composed relations $R_1 \circ \cdots \circ R_n$ and their admissibility. By Axiom 4, ontology cannot be reconstructed from dynamics. By Axiom 5, distinct relational structures may induce indistinguishable dynamic behavior. Since dynamics constrain neither the composition operator $\circ$ nor the admissible space determined by relations, architectural structure remains underdetermined by dynamics alone.

—

Theorem 2 (Coherence as Non-Empty Admissibility).

> *A composed relational structure $R_1 \circ \cdots \circ R_n$ is coherent if and only if it admits a non-empty set of admissible configurations.*

Proof sketch. By Axiom 2, architectural existence requires relations to compose without contradiction. In the relational algebra of ACS, admissibility is determined by the joint constraint imposed by composition. If no admissible configuration exists, the composed relations are incompatible and coherence fails. Conversely, the existence of at least one admissible configuration implies compatibility of all composed relations and hence architectural coherence.

—

Theorem 3 (Persistence Does Not Imply Architectural Existence).

> *Dynamic persistence of behavior does not imply architectural existence.*

Proof sketch. Architectural existence depends on the coherence of composed relations. A system may exhibit ongoing dynamics even when its relational structure $R_1 \circ \cdots \circ R_n$ is incompatible or violates architectural invariants. By Axiom 3, observation of persistent behavior cannot restore architectural truth once coherence is lost. Therefore, dynamic persistence is insufficient to establish architectural existence.

—

Corollary 1 (Functional Equivalence Is Ontologically Insufficient).

> *Functional or behavioral equivalence does not entail equivalence of relational architecture.*

Justification. Distinct relational structures may induce identical admissible dynamics. Since the composition operator $\circ$ and invariant preservation $\models$ are not uniquely determined by observed behavior, functional equivalence under observation underdetermines architectural structure.

—

Corollary 2 (Architectural Degradation May Be Latent).

> *Violation of architectural invariants may occur without immediate observable failure.*

Justification. Architectural degradation corresponds to failure of invariant preservation, $R \not\models I$. By Axiom 3, observation does not directly track architectural truth. Hence invariant violation may precede or remain hidden from observable dynamic breakdown.

5 Architecture and Dynamics

This section clarifies the ontological distinction between architecture and dynamics within ACS. While dynamics describe how systems evolve, architecture determines whether a system exists as a coherent entity at all. The distinction is not methodological but ontological: dynamics presuppose architecture, whereas architecture is not derivable from dynamics.

5.1 States, Transitions, and Architectural Space

Dynamics are commonly described in terms of states and transitions. A state represents a configuration of a system at a given moment, while transitions describe changes between such configurations. These notions are inherently temporal and presuppose a space within which change can occur.

Architecture defines this space. Through relational constraints, composition, closure, coherence, and invariants, architecture specifies the set of admissible states and transitions. Dynamics operate entirely within this admissible space; they do not define it. A trajectory may traverse admissible states, but the admissibility of those states is an architectural property, not a dynamic one.

5.2 Reversibility and Structural Underdetermination

Dynamic processes may be reversible. Given sufficient information about system dynamics, past states may be reconstructed from later states. Such reversibility, however, concerns state reconstruction rather than architectural reconstruction.

Architectural structure is not encoded uniquely in state trajectories. Multiple relational architectures may admit identical sets of states and transitions, or generate indistinguishable dynamic behavior. Reversibility of dynamics therefore does not imply reversibility of ontology. State reconstruction does not entail reconstruction of architectural structure.

5.3 Dynamics as Selection Within Admissibility

Within ACS, dynamics are understood as selection mechanisms. They select particular trajectories within the space of admissible configurations defined by architecture. Different dynamics may operate over the same architectural space, and identical dynamics may operate over distinct architectural spaces.

Because dynamics select rather than define admissibility, they cannot establish architectural truth. At most, dynamic behavior can be consistent or inconsistent with a given architecture. Consistency does not imply uniqueness, and inconsistency does not specify which architectural relations have failed. Dynamics thus provide evidence of compatibility, not determination.

5.4 Functional Equivalence and Ontological Distinction

Dynamic equivalence is often taken as evidence of structural similarity. If two systems exhibit indistinguishable behavior under observation, they are frequently treated as functionally equivalent. Within ACS, functional equivalence is recognized as a property of dynamics, not of ontology.

Ontological distinction persists even under perfect dynamic equivalence. Systems that realize identical admissible behaviors may differ in their relational architectures, invariants, or conditions of coherence. Functional correspondence therefore does not collapse architectural plurality. Architecture is defined by relational structure, not by behavioral correspondence.

5.5 Architectural Degradation and Dynamic Continuity

Architectural coherence may be lost without immediate dynamic collapse. A system may continue to exhibit behavior even after its relational structure has become inconsistent, incomplete, or externally dependent. Such behavior reflects dynamic continuity rather than architectural existence.

Within ACS, architectural degradation corresponds to violations of coherence, closure, or invariants. Once these conditions fail, the system no longer exists as a coherent architectural entity, regardless of ongoing dynamics. Dynamic persistence without architectural coherence characterizes processes rather than systems.

5.6 Ontological Priority of Architecture

The considerations above establish the ontological priority of architecture over dynamics. Architecture defines the conditions of existence and the space of admissibility. Dynamics describe motion within that space but cannot generate or recover architectural structure.

This priority entails a fundamental asymmetry. Architecture constrains dynamics, but dynamics do not determine architecture. Any account of systems that treats dynamics as ontologically primary implicitly presupposes an architecture it cannot itself justify. ACS makes this presupposition explicit and treats architecture as a first-order ontological concern.

5.7 Architectural Plurality and Non-Uniqueness

The axioms of ACS entail that architectural structure is not uniquely determined by behavior, dynamics, or observation. For a given set of constraints and admissibility conditions, multiple distinct relational architectures may exist. There is therefore no single correct or optimal architecture in an absolute sense.

Architectural reasoning within ACS does not aim at identifying a unique solution. Instead, it characterizes a space of admissible architectures, within which different structures may satisfy the same constraints to varying degrees. Effectiveness is thus relative to specified architectural conditions rather than derived from universal optimality.

Within this space, architectures may be compared with respect to how well they satisfy given constraints, preserve invariants, or support intended forms of stability. Optimization, where applicable, is therefore local and conditional. It operates over admissible architectures, not over architecture as such.

6 Relation to Foundational Frameworks

This section situates the Architecture of Complex Systems (ACS) within the broader research program from which it originates. ACS is not introduced as an isolated theoretical construct. It is developed in conceptual continuity with two foundational frameworks: the Philosophy of Discrete Being (FDB) and the Coherent Observational Epistemology (COE). The purpose of this section is not to restate these frameworks, but to clarify how ACS extends and specializes their core commitments.

6.1 Architecture and Discrete Being

FDB establishes an ontological framework in which existence is treated as fundamentally discrete. Entities exist not as continuous substances, but as localized, relationally constituted structures. Existence is therefore not a primitive given, but an outcome of specific conditions that allow a locality to persist as a coherent unit.

ACS inherits this ontological stance. Architectural existence is defined not by continuity, persistence, or dynamic activity, but by relational coherence and closure. A system exists architecturally only insofar as its relational structure satisfies the conditions of coherence specified in the ontological core of ACS. In this sense, architecture provides the structural criteria by which discrete being manifests at the level of complex systems.

While FDB addresses existence at the most fundamental ontological level, ACS operates at the level of structured multiplicities. It specifies how discrete entities may form higher-order coherent structures without appealing to continuity or emergent dynamics as primitive explanations. Architecture thus functions as the mediating layer between discrete being and complex organization.

6.2 Architecture and the Limits of Observation

COE develops an epistemological framework centered on the limits of observation. It emphasizes that knowledge is constrained by the conditions under which observation is possible, coherent, and meaningful. Observational data do not directly reveal ontological structure; they provide partial and context-dependent access to system behavior.

ACS incorporates this epistemological constraint into its ontological formulation. The axioms of ACS explicitly deny the reconstructibility of architectural structure from observation or dynamics. This denial is not an epistemic pessimism, but an ontological boundary condition. Architecture is defined as that which precedes and constrains observation, not as that which can be inferred from it.

By grounding architectural truth in relational structure rather than in observational adequacy, ACS aligns with the central thesis of COE. Observation is treated as secondary to ontology. The limits identified by COE are therefore not external restrictions imposed on ACS, but intrinsic consequences of architectural primacy.

6.3 Structural Continuity Across Frameworks

Together, FDB, COE, and ACS form a coherent ontological–epistemological progression. FDB specifies the conditions of discrete existence. COE specifies the conditions and limits of knowing. ACS specifies the conditions under which coherent structures exist as systems.

Each framework addresses a distinct level of analysis while remaining structurally consistent with the others. None is reducible to another, and none functions as a replacement. ACS does not reinterpret FDB or COE; it operationalizes their shared commitments at the level of architectural organization.

This continuity ensures that ACS is neither a domain-specific theory nor a detached abstraction. It is grounded in a unified research program that treats existence, knowledge, and architecture as mutually constrained but irreducible dimensions of systematic inquiry.

7 Conclusion and Outlook

This work has introduced the Architecture of Complex Systems (ACS) as an ontological framework for reasoning about systems as coherent architectures. Rather than describing behavior or modeling dynamics, ACS specifies the relational conditions under which systems exist, persist, and degrade as unified entities. Architecture is treated as ontologically prior to both dynamics and observation.

The core contribution of this paper is the formulation of a minimal ontological nucleus for architectural reasoning. By establishing relational primacy, coherence as a condition of existence, and the non-reconstructibility of ontology from dynamics, ACS delineates a clear boundary between architectural structure and system behavior. Relational algebra is employed as a working language to articulate admissibility, invariants, and architectural closure without reference to domain-specific realizations.

A central implication of ACS is that architectural truth cannot be inferred from observational adequacy or functional equivalence. Dynamics may be reversible, predictable, or stable, while architectural coherence may already be absent. Conversely, architectural integrity may persist independently of particular dynamic realizations. This asymmetry establishes architecture as a first-order ontological concern rather than a derivative descriptive layer.

The present work is intentionally foundational. It does not address specific domains, implementations, or empirical cases. Such applications are treated as extensions of ACS rather than as components of its core. Future work will develop domain-specific interpretations of ACS across technical, biological, social, and physical systems, while preserving the ontological commitments established here.

By fixing the ontological core of architectural existence, ACS provides a stable foundation for systematic inquiry into complex systems. It offers a unified language for distinguishing systems from processes, structure from behavior, and existence from observation. Subsequent developments may elaborate, refine, or extend this framework, but they do so on a foundation that is now explicitly defined.

Architectural Algebra. The Architecture of Complex Systems may be understood as an algebra of architectural existence. Rather than modeling behavior or simulating dynamics, ACS provides a minimal relational language for expressing the conditions under which systems exist as coherent architectures. Relations, their composition, admissibility, and invariant preservation constitute the algebraic core through which architectural truth is defined. In this sense, ACS is not a descriptive framework but a constructive one: it specifies what can exist architecturally before any consideration of how it behaves.

Architectural Non-Uniqueness. ACS does not posit the existence of a single correct or optimal architecture. For a given set of architectural constraints, multiple admissible architectures may exist. Architectural effectiveness is therefore evaluated relative to specified conditions and invariants, rather than derived from universal optimality.

References

[1] A. A. Nekludoff, "Coherent observational epistemology: Foundational principles, secondary principles, and axiomatic system," Zenodo, 2025. doi: `10.5281/zenodo.17632756` [Online]. Available: `https://doi.org/10.5281/zenodo.17632756`

[2] A. A. Nekludoff, "Philosophy of discrete being: Foundations and structural architecture," Zenodo, 2025. doi: `10.5281/zenodo.17690594` [Online]. Available: `https://doi.org/10.5281/zenodo.17690594`

A Notation and Conventions

This appendix summarizes the notational and conceptual conventions used throughout the paper. The purpose of this section is not to introduce additional formalism, but to provide a compact reference for the relational language employed in ACS.

Architectural Relations

Architectural relations are denoted by capital letters $R, R_1, R_2, \ldots$. A relation represents an architectural constraint linking entities or localities. Relations are ontological in nature: they specify conditions of admissibility rather than interactions, events, or observations.

Composition

The composition of relations is denoted by the operator $\circ$. For relations R_1 and R_2, the expression $R_1 \circ R_2$ denotes their joint applicability as a composed architectural constraint. Composition does not imply temporal order, causation, or execution. It expresses structural conjunction within an architectural context.

Locality and Restriction

Localities are denoted by L. The restriction of a relation R to a locality L is denoted by $R \upharpoonright L$. Restriction localizes architectural constraints and prevents relations from being treated as globally applicable abstractions.

Admissibility and Compatibility

Admissibility refers to the existence of configurations that satisfy a given relational structure. A relational composition is said to admit a configuration if the composed constraints are jointly satisfiable. Compatibility of relations is understood in terms of admissibility: incompatible relations admit no admissible configurations.

Invariants

Architectural invariants are denoted by I. An invariant expresses a condition that must hold across all admissible configurations of a system. Invariant preservation is denoted by $R \models I$, indicating that the relational structure R preserves the invariant I throughout its admissible space. Failure of invariant preservation, written $R \not\models I$, corresponds to architectural degradation.

Scope and Interpretation

All notation used in this work is interpreted ontologically rather than computationally. The operators and symbols introduced here do not describe algorithms, executions, or measurements. They serve as a compact language for expressing architectural existence, coherence, and degradation within the framework of ACS.