THE ONTOLOGY OF THE EFFECT PROPAGATION PROCESS

Marvin Hansen
marvin.hansen@gmail.com
Date: August 14, 2025

1 Ontology

2 The Ontology of the Effect Propagation Process

2.1 The Ontology of Being

The static ontology defines the concrete primitives that are the fundamental building blocks, the "matter" (hyle) and "form" (morphe) of the EPP, as defined by the Metaphysics of Being. Each primitive is a direct instantiation of the Monoidic Primitives and Isomorphic Recursive Composition, designed to represent distinct categories of existence within the EPP's universe. This section details these foundational entities, which, when combined, form the foundation upon which all dynamic processes and emergent phenomena are built.

2.1.1 The Causaloid: A Unit of Causality

Causality is fractal. A high-level cause (e.g., "economic recession") can be decomposed into a network of smaller, interacting causes (e.g., "inflation," "supply chain disruption," "interest rate hikes"), each of which can be further decomposed into smaller causes. Classical causality has always struggled with this reality because structural composition is fundamentally at odds with both the SCM representation and the causal DAG.

The EPP proposes to represent causality closer to its fractal nature: The Causaloid. In line with the EPP metaphysics, the Causaloid is a monoid.

The identity element of the monoid is the Singleton ‘Causaloid‘. This is the fundamental unit of causality. It has an id as identity and a causal function that captures the causal relationship. The form of the Causaloid is isomorphically recursive to enable uniform composition. The causaloid is isomorphic in the sense that a causaloid can have different types and yet share the shape of a causaloid. For example, a causaloid can be:

1. A singleton causaloid that is a single cause.
2. A collection of causaloids.
3. A graph in which each node is another causaloid.

Applying the concept of isomorphically recursive composition means that any number of causaloids of different types uniformly compose. A complex causal graph can be encapsulated into a single Causaloid, which then becomes a node in another causal graph that itself is part of a causal collection. This creates a mechanism where:

- Complexity is manageable
- Scalability is inherent
- Abstraction is first-class

Complexity is manageable because each causaloid can be built and tested in isolation, or as part of a specific subgraph. The mechanism scales from a singular unit to larger collections up to complex graphs. While the system complexity scales, the conceptual overhead remains constant because a single concept, the causaloid, scales from simple to complex, from small to large. Abstraction is first-class because causal reasoning happens uniformly regardless of the underlying complexity. The reasoning mechanism remains the same regardless of whether a causaloid is a singleton, a collection, or a graph.

The monodic binary operation for composition results from the fact that causaloids are isomorphic, which allows to combine a causal collection into a new singleton causaloid. Likewise, an arbitrarily complex causal graph composes uniformly any number of causaloids as its nodes and is in itself a causaloid that composes with other causaloids.

Isomorphic Recursive Composition for causality defines an ontology where the distinction between "part" and "whole" is fluid. Every "whole" (a composite Causaloid) can become a "part" in a larger whole without ever changing its fundamental nature as a Unit of Causality.

2.1.2 The Contextoid: A Unit of Context

The Contextoid represents the monoidic primitive of state, an atomic, non-recursive, and identifiable unit of factual information. The EPP ontology makes a strict distinction between these two primitives: Causaloids are recursive and represent reasoning; Contextoids are isomorphic and represent the ground truth upon which reasoning operates. The ontology of the Contextoid does not enforce a single, fixed representation for concepts like "space" or "time." Instead, it defines abstract categories:

- Datoid - A monoidic unit of data-like context.
- Spaceoid - A monoidic unit of space-like context.
- SpaceTempoid - A monoidic unit of spacetime-like context.
- Symboid - A monoidic unit of symbol-like context.
- Tempoid - A monoidic unit of time-like context.

The categorical isomorphism leads to the most critical design principle of this primitive: Contextoids are structurally non-recursive. A Tempoid cannot contain another Tempoid; a Spaceoid cannot contain another Spaceoid. The structural prohibition of recursion guarantees the logical consistency of the Context. It makes it impossible to construct a paradoxical state, such as a time loop where a moment in time is defined as preceding itself. By ensuring the factual bedrock is non-paradoxical and acyclic, the EPP provides the stable, non-contradictory ground truth required for sound, higher-level reasoning to operate.

Conversely, the same categorical isomorphism enables heterogeneous composition within each category. For example, the Spaceoid category can represent a point in a flat, Euclidean space alongside another representing a point in a non-Euclidean GeoSpace. From the perspective of the ontology, both are valid "spatial facts," i.e., instances of a Spaceoid and thus uniformly treated as space-like. This categorical isomorphism enables the EPP to model heterogeneous systems where different parts of reality demand different mathematical representations of space, time, spacetime, symbol, and data. However, the absence of isomorphic recursion deprives the contextoid of its structure.

2.1.3 The Context: A structure for Contextoids

The Context is the structure that gives meaning through relations to the Contextoids. In the EPP, the Context is a first-class ontological entity: a hypergraph structure that holds all Contextoid primitives and the explicit, typed relationships between them. It is partially monodic in the sense that it has an identity, but unlike other primitives, it has no monoidic composition hence its name “Context” reflects that it is not monoidic and thus does not compose. The deliberate choice roots in the fact that, while its elements, the contextoids, are structurally isomorphic and compose, the engulfing structure, the context, does not compose to prevent the complexity that arises from merging graphs, but more importantly, to prevent incorrect states in this critical structure. Despite this decision, the context, however, is neither singular nor absolute.

Static vs Dynamic Context

A static context is established upfront and is structurally assumed to remain invariant during its lifetime. The utility comes from encoding static knowledge, for example, the ICD-10 medical ontology that is standardized for a given version. A static representation for each version of the ICD ensures there is no mixing of different standards.

A dynamic context is one whose structure is dynamically built and modified during its lifetime. The utility applies to situations where context is fluent but structurally known. For example, a monitoring system that receives data feeds from maintenance drones has to add contextoids for drones coming online and streaming data and, likewise, remove contextoids for drones that get out of range and stop the data stream. The dynamic context can only add, modify, or remove contextoids of known types to ensure strict operational safety.

Single vs Multiple Contexts

The EPP ontology explicitly supports multiple frames of reference via multiple contexts. A causal model can be linked to a primary context representing the current, observable reality, while also having access to any number of additional contexts. These auxiliary contexts can represent simulated worlds, historical states, counterfactual scenarios, or backup data sources for real-time data feeds.

Contextual Counterfactuals

This capacity for multiple contexts establishes the foundation for relativistic contextual counterfactual analysis. It allows the system to ask "what if" questions by creating hypothetical realities (a new extra context), modifying specific Contextoids within them (e.g., "what if temperature were 5

2.1.4 The Evidence: A Unit of Facts

Evidence represents a specific monoidic fact presented to a Causaloid for evaluation that may originate from outside the system, i.e., from a sensor reading, is extracted from a contextoid, or is derived from a previous chain of reasoning from a causaloid. A causaloid uses Evidence as input and the context for supporting data used during the analysis. The output of a Causaloid can then be directly verified against the specific Evidence it received, which is fundamental for explainability in complex systems.

Evidence is a generalized isomorphically recursive container designed to support unified reasoning across multiple modalities:

- Deterministic: Boolean values ("true/false").
- Numerical values: Numbers (e.g., sensor readings).
- Probabilistic values (e.g., confidence scores, likelihoods).
- Contextual Links (a Contextoid within a Context), enabling the Causaloid to access complex, structured, and non-numerical facts.

Evidence is recursive and may contain:

- Maps of other Evidence primitives
- Graphs of Evidence primitives enable a Causaloid to access complex, relational data.

The multi-modal Evidence enables the EPP to reason over the full spectrum of information found in complex systems.

2.1.5 The Propagating Effect: A Unit of Influence

The output of a causal evaluation is the PropagatingEffect, a monoidic primitive of influence. The PropagatingEffect is the operational heart of the "Effect Propagation Process" itself, a unit of influence that travels through the causal graph, from one Causaloid to the next, driving the continuous effect propagation process. The PropagatingEffect is a unified inference outcome across different modalities. It is isomorphic in that it can represent:

- Deterministic effect: A definitive boolean outcome ("true/false").
- Probabilistic effect: A quantitative outcome, such as a probability score or an estimate.
- Contextual Link: A reference to a specific Contextoid within a Context.

For the Contextual Link, a causaloid writes its reasoning outcome into a contextoid and then propagates its "effect" as a Contextual link to direct the next Causaloid to use the structured information in the contextoid for further analysis. This Contextoid then becomes the Evidence for the subsequent step in the reasoning chain, enabling dynamic, data-driven causal pathways for non-numerical representation as required, for example, for causal symbolic reasoning enabled by the symboid contextoid type.

2.2 The Ontology of Becoming: Dynamics

The ontology of dynamics in the EPP is governed by the metaphysical principle of contextual relativity. Contextual means that the significance of a monoidic primitive is derived from its relation to its context. Contextual relativity means that state is an emergent property arising relative to its context with the implication that the same object may have different states when used in different contexts, but, more profoundly, the same object may alter its state when context itself is relativistically altered and propagates its contextual adjustment to all connected monoidic primitives. Contextual Relativity is expressed in the EPP in two ways:

First, a Causaloid's reasoning is relative to the 'Context' it is evaluated against. The EPP ontology explicitly supports multiple frames of reference for each causaloid. A Causaloid asking "Is the pressure critical?" can return true when evaluated against a Context representing a high-altitude environment and false when evaluated against one representing sea level. The causal logic is the same, but the truth it produces is relative to the frame of reference. This is very much in line with the observer principle in the general theory of relativity.

Second, the state of a Contextoid itself is subject to relativistic forces imbued upon its engulfing fabric. For example, a spacetime contextoid may need to adjust its temporal value for gravity-induced time dilation. A quaternion contextoid may need to adjust its rotation value relative to incoming sensor data. Therefore, the EPP ontology defines two critical operations to handle contextual relativity via two operations:

1. Adjust
2. Update

Adjust means an existing value is adjusted for relativistic effects or corrected for sensor drift. The core property of an adjustment operation is to take a correction value, say an offset, and apply it to the existing value.

Update means an existing value is replaced with a new value, for example, a new sensor reading. The core property of an update operation is to take a data value and replace the current one with the new one.

These two operations, adjust and update, determine the difference between a strictly static contextoid and a dynamic contextoid. A static one is assumed to be invariant throughout its lifetime, for example by holding a set of immutable facts, i.e., a thermal threshold. A dynamic contextoid is one that gets either adjusted or updated, for example when new sensor reading becomes available. A simple causaloid then reads the thermal threshold from one contextoid, and reads the current thermal sensor data from a dynamic contextoid, and determines if it is getting closer to the thermal limit. Critically, when transferring the model to a different operational environment that requires a different thermal threshold, the replacement of one contextoid is sufficient to ensure the correct functionality of the modeled thermal system.

2.3 The Ontology of Emergence

The ontology of Emergence is captured in the mechanism of the Generative Process, a four-stage command-execution cycle that transforms external stimuli or internal states into structural modifications of the EPP itself.

1. The Generative Trigger
2. The Generative Command
3. The Generative Process
4. The Generative Outcome

This four-stage process applies to both first-order emergence and higher-order emergence because of its recursive design. Ontologically, each step results in a specific and verifiable object, which lays the foundation for a principled implementation at a later stage.

2.3.1 The Generative Trigger

Emergence begins with a Generative Trigger. This primitive represents the initial impetus for change, signaling to the EPP that a modification of its internal structure becomes necessary. Triggers can originate from external stimuli, the passage of time, or an explicit external command. A generative trigger can arise from internal states, such as the detection of an anomaly, a deviation from a desired goal, or from a change in external state such as the detection of a fundamental change in the environment, i.e., transiting from day to night. The Generative Trigger acts as the catalyst to initiate the generative process.

2.3.2 The Generative Command

Upon receiving a Generative Trigger, a Generative Command is constructed. This primitive is a declarative, verifiable blueprint of a desired action or structural modification. It is the formal expression of the EPP's intent to alter its own substance. Generative Commands are explicit instructions for change, such as the creation, update, or deletion of a Causaloid or Contextoid, the modification of contextual relationships by adding or removing edges in the hypergraph.

The Generative Command is both isomorphic and recursive. It is isomorphic in that it unifies a diverse set of possible actions under a single primitive type. These actions include, but are not limited to:

- No operation. *NoOp*:
- Commands for managing causal logic: *CreateCausaloid*, *UpdateCausaloid*, *DeleteCausaloid*
- Commands for managing contexts: *CreateBaseContext*, *CreateExtraContext*, *UpdateContext*, *DeleteContext*
- Commands for managing facts within a context: *AddContextoidToContext*, *UpdateContextoidInContext*, *DeleteContextoidFromContext*
- A user-defined command for higher-order emergence: *Evolve*.

The Generative Command is recursive through its Composite variant. This variant allows a single Generative Command to contain an ordered sequence (a "stack") of other Generative Commands. This recursive capability enables the EPP to express complex, multi-step transformations as a single, atomic unit of intent. For example, a single Composite command could specify: "add root node, then add Causaloid A, then add an edge between root and A." This ensures that even intricate sequences of operations are treated as a coherent, atomic transaction, maintaining the integrity and verifiability of the system's emergence.

The Evolve command is one notable outlier in that it is not an atomic command. Instead, Evolve is a meta-command; it is an instruction to replace the mechanism that generates commands itself. By definition, Evolve is no longer guaranteed to be deterministic. If the choice of what to evolve into is not itself deterministically derived, then the system's future behavior becomes non-deterministic and potentially unpredictable. Even if the new command is deterministic, the decision to evolve and how to evolve represents a point where the system's fundamental archê kai aitia of change is altered and it is conceptually unclear what this may entail in practice.

2.3.3 The Generative Process

The Generative Process is the operation responsible for actuating the Generative Command. It interprets the command and translates it into concrete, structural modifications within the EPP's ontology.

This stage represents the materialization of the intended change, where new Causaloids are instantiated, Context graphs are reconfigured, or Contextoids are added, modified, or removed. The Generative Process transforms the abstract intent into tangible alterations of the EPP's substance.

The Generative Process is designed to be robust and auditable. It processes each Generative Command deterministically, ensuring that the system's state transitions are predictable given a specific command. Its primary function is to ensure the integrity of the EPP's internal structure during the process of self-modification.

2.3.4 The Generative Outcome

The generative outcome is a primitive that holds both the modified or generated artifact and relevant metadata about the process itself. It attests to the “what happened” during the process in the form of a log detailing every state transition and asserts whether the outcome is in accordance with the stated intention of the generative command. This explicit record of the system's self-modification means that the EPP maintains a complete and verifiable history of its own evolution; a crucial step for auditability, debugging, and the continuous learning process, providing the ground truth for subsequent development of emergence in complex dynamic systems.

2.4 Discussion

The ontology of the Effect Propagation Process addresses the limitations of classical causal models in dynamic, complex systems. The EPP's ontology deeply integrates a set of primitives: the Causaloid as a recursive unit of logic, the Contextoid as a non-recursive unit of fact, and the Context as a dynamic, multi-frame relational environment. These primitives, combined with the multi-modal Evidence and PropagatingEffect, form a robust foundation for expressing causality not as a static, linear chain, but as a continuous process of effect propagation.

The EPP's explicit ontological primitives and its four-stage Generative Process provide a unique computational platform for exploring the nature of emergence itself. While the principles of Dynamic State Mutation and Dynamic Structural Evolution offer clear pathways for predictable adaptation, the concept of Dynamic Co-Emergence, through the Evolve command, opens a frontier for investigating self-referential self-modification of dynamic systems.

The Evolve command, as defined within the EPP's ontology, represents the system's capacity to fundamentally alter its own generative principles. This capability, while theoretically profound, introduces complex questions regarding predictability, control, and safety. The EPP's transparent, auditable architecture where every Generative Command and Execution Result is explicit offers an unprecedented opportunity for empirical investigation into these phenomena.

By designing and observing systems that engage in self-referential self-modification, practitioners can gain firsthand insights into the dynamics of emergent behavior, the challenges of maintaining verifiability in adaptive systems, and the potential pathways towards engineering truly autonomous and self-adapting dynamic systems. This area represents a rich vein for future research, where the EPP's foundational principles can be tested and expanded through direct computational experimentation.

3 Conclusion

The Effect Propagation Process (EPP) framework rethinks causality as a continuous transfer of effects originating from a a Euclidean or non-Euclidean environment that might be subject to relativistic effects. This paper has articulated the EPP as a novel computational ontology, grounded in a relational philosophy that directly addresses the limitations of classical causal models in dynamic, complex systems.

The EPP's core metaphysical principles, Monoidic Primitives, Isomorphic Recursive Composition, and Contextual Relativity, together form a robust framework for understanding the nature of being and becoming in a coherent and computable way.

The EPP's unique philosophical grounded design enables capabilities previously intractable for traditional methods. Its principle of Contextual Relativity allows for real-time, relativistic counterfactual analysis, where a system can explore multiple hypothetical futures by dynamically altering its frames of reference. This capacity for "what-if" reasoning, executed at scale, transforms autonomous decision-making from rigid, pre-scripted responses to adaptive, consequence-aware optimization.

The EPP ontology offers a unified philosophical language of causality that handles new challenges, remains transparent and verifiable, and provides the essential architectural primitives for dynamic emergent causality even in relativistic contexts.