Bliss: A Data–Action–Binding Semantic Model

Mayukh Chakraborty
kismotechindia@gmail.com

Abstract

This is a draft paper.

Modern programming languages routinely attach meaning, behavior, and guarantees to values implicitly. As systems grow, this implicit semantic coupling becomes a source of ambiguity, refactoring fragility, and accidental abstraction. This paper proposes a semantic model that enforces explicit semantic commitment by separating representation, behavior, and meaning into three disjoint domains: Data, Actions, and Bindings.

1 Introduction

Modern programming languages routinely attach meaning, behavior, and guarantees to values implicitly. As systems grow, this implicit semantic coupling becomes a source of ambiguity, refactoring fragility, and accidental abstraction. Developers are often required to reason about behavior that is not locally visible and about meaning that was never explicitly chosen.

This paper proposes a semantic model that enforces **explicit semantic commitment**. Instead of inferring meaning from structure, context, or usage, meaning is introduced only through explicit operations and is mechanically enforced. The model is intentionally pragmatic and explanatory in nature, following the spirit of system-level white papers rather than academic formalism.

The central principle guiding this work is simple:

Meaning is chosen, never inferred.

2 Semantic Model Overview

In this paper, *semantics* refers to the set of operations that are valid on a value and the guarantees those operations provide. Semantics does not

include representation, syntax, or implementation strategy.

A *value* is a runtime entity on which operations may be defined and applied. Depending on its semantic domain, a value may carry representation, behavior, and semantic guarantees.

The proposed model separates semantics into three distinct and disjoint domains:

1. **Data** — representation without meaning
2. **Actions** — behavior without meaning
3. **Bindings** — explicit semantic commitment

Transitions between these domains are explicit. No domain is implicitly derived from another, and no semantic information is inferred across boundaries.

3 Data: Open Semantics

Data represents raw memory and layout. A data value provides guarantees only about representation, such as size and structure. It provides no guarantees about behavior, interpretation, or correctness.

Because data carries no inherent meaning, it exists in an *open semantic state*. Operations that reinterpret data under different behavioral or semantic views are permitted, as no semantic contract is being violated.

Data forms the foundation of the model. All higher semantic constructs ultimately rely on data, but data itself remains free of behavioral and semantic assumptions.

4 Actions: Behavior Without Meaning

Actions define behavioral capabilities. An action specifies a set of operations that may be performed, without prescribing what those operations mean in a broader semantic sense.

Actions do not define storage, do not have instances, and do not impose interpretation. Two values implementing the same action may have entirely different representations and meanings.

By separating behavior from meaning, actions enable behavioral polymorphism without semantic commitment. They answer the question *"what can be done?"* without answering *"what kind of thing is this?"*

5 Bindings: Closed Semantics

A binding creates a semantic type by explicitly attaching an action to a data type. This operation commits the data to a specific behavioral interpretation and establishes a *closed semantic contract*.

Once bound, a value's meaning is fixed. Operations outside the bound semantic universe are forbidden, and reinterpretation is disallowed. This closure ensures that semantic guarantees remain stable and local.

Bindings are nominal rather than structural. Two bindings over the same data and action are distinct semantic types unless explicitly unified. This prevents accidental equivalence and enforces explicit semantic intent.

6 Semantic Boundaries and Transitions

Operations and call boundaries define semantic transitions. Each transition explicitly determines which semantic information is preserved across it.

Some boundaries erase semantics entirely, exposing only raw data. Others preserve full semantic meaning, while some preserve only behavioral guarantees. These transitions are explicit and mechanically enforced.

By making semantic preservation and erasure explicit, the model prevents unintended semantic leakage and clarifies which guarantees remain valid across boundaries.

7 Composition Rules

The model permits limited and explicit composition. Actions may be composed at the value level to expose multiple behaviors simultaneously, without creating new semantic types.

Bindings do not compose. Each binding introduces a single, closed semantic commitment. This restriction is intentional and prevents the emergence of ambiguous or widened semantics.

Composition rules exist to ensure that the semantic universe of a value remains explicit, finite, and locally understandable.

8 Implications of the Model

Several consequences follow directly from this model. Semantic guarantees are always local and explicit, and refactoring affects only the semantic types that are directly modified.

Because data is defined independently of meaning, systems naturally evolve in a data-first manner. Semantic commitments can be introduced, revised, or replaced without destabilizing underlying representations.

The absence of global coherence rules further ensures that semantic reasoning remains localized and predictable.

9 Benefits

The explicit separation of representation, behavior, and meaning simplifies reasoning about large systems. Developers can determine what operations are valid on a value without relying on external context or implicit rules.

The model improves refactorability, supports long-lived systems, and enables tooling to reason mechanically about semantic guarantees. Accidental abstraction and implicit coupling are structurally prevented.

10 Restrictions

The model intentionally forbids several patterns. Semantics cannot be widened implicitly, and meaning cannot be inferred from structure or usage.

Bindings cannot compose, and reinterpretation is forbidden once semantics are closed. These restrictions are design choices that prioritize explicitness and predictability over convenience.

11 Limitations

The model does not attempt to enforce semantic correctness beyond explicit guarantees. Unsafe reinterpretation of raw data remains possible and requires programmer discipline.

Certain expressive patterns may require additional boilerplate or explicit modeling. These limitations are accepted tradeoffs rather than oversights and represent areas for future exploration.

12 Conclusion

This paper presented a pragmatic semantic model that separates representation, behavior, and meaning into distinct, explicitly connected domains. By enforcing deliberate semantic commitment, the model eliminates implicit meaning and improves local reasoning.

The guiding principle of this work—*meaning is chosen, never inferred*—underpins all aspects of the design. Future refinements may formalize or extend the model, but the core objective remains the same: to make semantics explicit, local, and mechanically enforceable.

References

- (Add citations as needed.)