UST v1.0 — Universal Semantic Token Model

Robert Hansen

November 2025

Abstract

UST v1.0 defines a universal, domain-agnostic semantic token model for deterministic AI systems. The model provides a typed, extensible structure for representing meaning, intent, constraints, and routing data in a machine-verifiable format. UST enables AI runtimes and orchestration engines to operate on meaning rather than raw natural language, enabling predictable behavior, reproducible flows, and scalable multi-agent coordination.

1 Introduction

Modern AI systems rely primarily on natural-language prompts. While flexible, prompts lack explicit structure, type safety, and deterministic interpretation. As multi-agent systems scale, prompt-driven workflows break or drift.

The Universal Semantic Token Model (UST) establishes a unified semantic substrate that captures:

- domain-neutral meaning,
- system intents and directives,
- constraints and guarantees,
- routing and control metadata.

UST serves as the base layer for deterministic execution, state management, and multi-agent orchestration.

2 Design Principles

UST is guided by five principles:

2.1 Universality

The schema applies across domains without modification.

2.2 Determinism

Tokens support reproducible interpretation regardless of runtime context.

2.3 Extensibility

Families and types may be added without breaking existing systems.

2.4 Clarity

Each token has a single, unambiguous meaning.

2.5 Governance Compatibility

Tokens support validation, auditing, and external governance.

3 Token Structure

A UST token is defined as:

```
Token {
    id: UniqueIdentifier
    family: TokenFamily
    type: TokenType
    version: Version
    payload: TypedPayload
    constraints: ConstraintSet
    metadata: Metadata
}
```

3.1 Unique Identifier

A stable hash across family, type, version, and payload signature.

3.2 Token Family

A broad semantic category, e.g., Semantic, Teleo, Trade.

3.3 Token Type

A subtype representing a specific semantic role.

3.4 Version

Semantic version of the token type.

3.5 Typed Payload

Structured content with type guarantees.

3.6 Constraints

Optional validation, safety, or behavioral limits.

3.7 Metadata

Contextual information including timestamps or provenance.

4 Token Families in v1.0

Three families are standardized:

4.1 Semantic Tokens

Represent descriptive information, concepts, or relationships.

4.2 Teleo Tokens

Represent goals, objectives, or directed intent.

4.3 Trade Tokens

Represent value exchange, negotiation, or commitments.

5 Deterministic Interpretation

Deterministic behavior emerges from:

- typed payload specifications,
- versioned schemas,
- explicit constraints,
- stable identifiers,
- predictable validation rules.

Agents must not infer meaning beyond the token contents.

6 Validation Rules

Each token must satisfy:

1. Schema validity
2. Type-family consistency
3. Payload-type correctness
4. Constraint consistency
5. Version registration

Invalid tokens must not be interpreted.

7 Extending UST

Extensions require:

- unique namespaces,
- defined payload schemas,
- defined validation rules,
- proof of determinism.

Backward-compatible additions do not require major version changes.

8 Integration with Orchestration Engines

UST integrates with deterministic planners and multi-agent systems. Tokens support:

- execution planning,
- state transitions,
- constraint enforcement,
- agent coordination,
- safe routing.

9 Example Token

```
Token {
    id: "sem_v1_concept_abc123",
    family: "Semantic",
    type: "Concept",
    version: "1.0",
    payload: {
        label: "Pipeline",
        attributes: ["Deterministic", "Auditable"]
    },
    constraints: {
        readonly: true
    },
    metadata: {
        timestamp: "2025-11-25T10:04Z"
    }
}
```

10 Future Directions

Potential areas include:

- temporal semantics,
- probabilistic payloads,
- multi-token bindings,
- domain expansions.

11 Conclusion

UST v1.0 provides a universal semantic representation suitable for deterministic AI systems, forming the semantic backbone for scalable, auditable, and reproducible multi-agent workflows.