NEXSYS CONTINUUM

Persistent Cross-Session State as an Extension of the SCIOUS / NEXSYS / EDIOS Architecture

1. Overview

This extension is defined within the existing four-layer hierarchy of the provisional filing (U.S. Provisional Patent Application No. 64/140,326): SCIOUS (the Computational Awareness paradigm), NEXSYS (the underlying Computational Intelligence Platform), EDIOS (the governed enterprise state and Decision Object layer built on NEXSYS), and, introduced here, Continuum (the persistence and resumption mechanism that extends EDIOS-governed state across sessions and entities).

This continuity principle is not new to this extension. The as-filed provisional package for Application No. 64/140,326 includes the SCIOUS and NEXSYS papers, which establish, as a foundational principle of the paradigm, that “state precedes computation” and that enterprise information is transformed into “a persistent Enterprise Computational State that serves as the substrate for intelligent computation.” NexSys Continuum formalizes and extends this already-disclosed persistent-state principle to the session, user, and process level; it does not introduce continuity as a new concept.
EDIOS already defines the governed state model: the Canonical Enterprise Model, the Enterprise Twin, Decision Objects, and the Decision Governance and Orchestration Engine that governs them. NexSys Continuum does not introduce a competing or parallel object model. It extends the existing EDIOS governed-state and Decision Object layer to a new class of tracked entity — the individual session, user, or process — and adds the specific mechanism by which that entity's EDIOS-governed state is resumed at the start of a new session rather than reconstructed from nothing.

This is presented as a continuation of the existing provisional filing, applying claimed EDIOS mechanisms — the Canonical Model, Decision Objects, and the Decision Governance and Orchestration Engine — to this new entity class, rather than introducing a separate claim architecture.

2. Core Constructs: EDIOS Decision Object Typing and the Continuum State Object

2.1 Decision Object Typing (object_class)

Within the existing EDIOS Decision Object construct, this extension specifies a typing attribute, object_class, that identifies the semantic character of a given Decision Object. This is a refinement of the existing Decision Object, not a new base type: every object described below remains a Decision Object within EDIOS's governed state model, carrying EDIOS's existing properties for provenance, temporal validity, governance, relationships, lifecycle, and significance. The object_class attribute distinguishes at least the following:

- DECISION — a made determination with downstream effect
- CONSTRAINT — a bounding rule or limitation the entity operates within
- COMMITMENT — a directive or binding forward obligation
- PREFERENCE — a descriptive, lower-stakes standing characteristic of the entity

- ASSERTION — a general stated fact about the entity not otherwise classified

Typing by object_class allows EDIOS's existing persistence, versioning, lineage, authorization, and audit mechanism to govern all Decision Objects uniformly, while still permitting subtype-specific properties (for example, a COMMITMENT may carry expiration or fulfillment status not applicable to a PREFERENCE), without requiring separate, parallel object systems for each semantic category.

2.2 Continuum State Object (CSO)

A Continuum State Object (CSO) is a governed state-resumption construct within EDIOS. It is not a competing or parallel container to the Decision Object — it is a reference structure that identifies the set of EDIOS Decision Objects and Deltas relevant to a given tracked entity (a session, user, or process) at a point in time, together with reference metadata linking it to any related enterprise-level or other-entity CSOs. The CSO allows an entity's EDIOS-governed state to be resumed as a whole at the start of a session, rather than reconstructed object-by-object. Because the CSO references Decision Objects generically by their existing EDIOS governance properties, its structure does not require redefinition as new object_class values are introduced.

2.3 Governed State Resumption

The novel mechanism is not that the system "remembers" prior interactions, but that it performs a structured, governed state resumption process at the start of each new session, built around the Continuum State Object:

- On session initialization, the system queries the persistent state store for the entity's current Continuum State Object.
- The State-Delta-Decision cycle resumes from the EDIOS Decision Objects and Deltas referenced within that Continuum State Object, rather than beginning from a null state.
- The Continuum State Object itself, and all Decision Objects it references, remain subject to EDIOS's existing governance, access, and audit controls — Continuum introduces no separate governance mechanism.

4. Signal Filtering and Classification: What Persists

Not every interaction constitutes a state change worth persisting. Continuum reuses the existing Enterprise Signals mechanism to filter session content, so that only interactions crossing a significance threshold are written as EDIOS Decision Objects and incorporated (by reference) into the entity's Continuum State Object. A classification step assigns each qualifying signal its object_class (DECISION, CONSTRAINT, COMMITMENT, PREFERENCE, or ASSERTION) based on the semantic character of the statement — for example, a stated standing characteristic ("prefers concise summaries") is classified as a PREFERENCE, while a stated forward-looking directive ("use this format for all future materials") is classified as a COMMITMENT. Routine or transient exchanges are not persisted and do not update the CSO. This keeps Continuum's state store aligned with the same signal-detection logic already claimed for enterprise-level monitoring, extended with object-level classification, rather than introducing a wholly separate filtering scheme.

5. Multi-Entity Governance: Personal and Enterprise State

Where an individual's Continuum State Object intersects with an enterprise's Continuum State Object — for example, an employee using the system both in a personal capacity and on behalf of an organization — the two are maintained as separate, independently governed CSOs, linked by reference rather than merged. This preserves distinct access and audit rules for personal versus enterprise Decision Objects while still allowing traceability between the two where the entity relationship requires it.

6. Retention and Expiration

Persisted EDIOS Decision Objects are retained under a policy aligned with e-discovery and legal-hold standards, consistent with the Decision Governance and Orchestration Engine's existing compliance role. A baseline retention period (e.g., seven years) applies by default, with legal hold capable of extending retention as required, and governed deletion or expiration applying otherwise. Retention policy may vary by object_class where warranted — for example, COMMITMENT and CONSTRAINT objects carrying binding or compliance significance may merit stricter retention than PREFERENCE objects.

7. Relationship to Future Embodiment (Roadmap)

Continuum's persistent, governed state architecture is also intended, as a longer-term roadmap item outside the scope of the current filing, to serve as the integration substrate for a future physical embodiment division (working name: NexSys Robotics / NexSys Embodiment). In this model, third-party conversational AI systems would interface with physical hardware through the entity's Continuum State Object in a model-agnostic manner, rather than the platform being tied to a single AI provider.