Core Emotion Framework (CEF): Technical Specification 3 (TS-3)

Computational Specification

Canonical Architecture-Level Document — Version 1.0

Jamel Bulgaria

ORCID: https://orcid.org/0009-0007-5269-5739
Affiliation: OptimizeYourCapabilities.com
Contact: jamebulgaria@gmail.com
Version: 1.0
Date: 2025-12-29

Abstract

The Core Emotion Framework (CEF) Technical Specification 3 (TS-3) defines the canonical computational architecture of the CEF. Whereas TS-1 establishes the operational mechanics of centers, processes, operators, and activation dynamics, and TS-2 defines the empirical validation architecture, TS-3 specifies the computational structures, update rules, matrix operations, and simulation cycles required for implementing the CEF in algorithmic and computational environments. TS-3 is an architecture-level document: it defines the formal computational rules and constraints governing all CEF-based simulations and models, without prescribing programming languages, code, or applied examples.

---

1. Purpose and Scope

1.1 Purpose

TS-3 establishes the computational specification of the CEF. It defines:

- computational representations of operators, processes, and centers
- vector and matrix structures for activation, modulation, and directionality
- update rules governing state transitions
- simulation cycle definitions and convergence criteria
- computational models for fusion, overflow, and identity preservation
- constraints ensuring fidelity to the canonical architecture

TS-3 translates the architecture defined in TS-1 and the validation logic defined in TS-2 into a computationally implementable form suitable for modeling, simulation, and algorithmic analysis.

1.2 Scope

TS-3 is an architecture-level computational specification. It defines:

- canonical data structures
- update functions
- matrix operations
- stability and boundary constraints
- computational rules for directionality, fusion, and overflow

TS-3 does **not** include:

- programming language implementations
- software engineering patterns
- code examples
- applied simulations or case studies

TS-3 provides the computational foundation upon which all CEF-based simulations, models, and algorithmic systems must be built. It is subordinate to TS-1 and TS-2 and must be interpreted in accordance with their definitions and constraints.

---

2. Computational Representation

2.1 Operator Representation

Each operator $O_{c,p}$ is represented as:

- a scalar activation value
- bounded within defined limits
- updated according to operator-level update rules

Deciding is represented as a constant-activation operator:
its activation is binary (engaged or not) and does not scale in magnitude.

2.2 Center Representation

Each center is represented as a 3-dimensional vector:

C = [C_Head, C_Heart, C_Gut]

Center activation is computed from operator activations using aggregation rules defined in TS-1.

2.3 Process Vector

The process vector is a 10-dimensional vector:

$P = [p_1, p_2, \ldots, p_{10}]$

Each element corresponds to a core emotional process.

2.4 Combined State Representation

The full emotional state is represented as:

- a 10-dimensional process vector
- a 3-dimensional center vector
- an optional concatenated state vector

Dimensionality and ordering must remain consistent across implementations.

---

3. Matrix Structures

3.1 Center Activation Matrix (3×3)

$A\_C[i, j]$ = influence of center i on center j

Constraints:

- non-negativity
- no zero rows
- full bidirectionality

3.2 Process Activation Matrix (10×10)

$A\_P[i, j] =$ influence of process i on process j

Constraints:

- zero entries for forbidden transitions
- symmetry only where reciprocity is defined

3.3 Operator Activation Matrix (30×30)

$A\_O[(c, p), (c', p')] =$ influence of $O\_\{c, p\}$ on $O\_\{c', p'\}$

Constraints:

- identity preservation
- structural boundaries
- no operator migration

3.4 Matrix Normalization

Matrices may require:

- scaling
- clipping
- normalization

to maintain stability and prevent divergence.

4. Update Rules

4.1 Operator Update Function

Operator activations update according to:

$O\_\{t + 1\} = f\_O(O\_t, A\_O)$

4.2 Center Update Function

Center activations update according to:

$C\_\{t + 1\} = f\_C(P\_t, A\_C)$

4.3 Process Update Function

Process activations update according to:

$P\_\{t + 1\} = f\_P(P\_t, A\_P)$

4.4 Combined Update Function

The full system update is:

$S\_\{t + 1\} = f(S\_t, A\_C, A\_P, A\_O)$

4.5 Stability and Convergence

A computational update is stable when:

- activation values remain within bounds
- transitions converge
- no oscillatory or divergent patterns emerge
- no chronic fusion is induced by the update rules

5. Simulation Cycle

5.1 Initialization

Simulations must define:

- initial operator activations
- initial center activations
- boundary conditions

5.2 Iterative Update

Simulations proceed through iterative updates:

- synchronous or asynchronous
- fixed or adaptive step size

5.3 Convergence Detection

Convergence is detected when:

- activation changes fall below a threshold
- no further structural transitions occur

5.4 Logging and Output

Simulations must record:

- state vector history
- activation trajectories
- fusion and overflow markers

---

6. Computational Modeling of Fusion and Overflow

6.1 Fusion Modeling

Fusion is modeled as:

- temporary cross-center modulation
- without operator migration
- without identity alteration

6.2 Chronic Fusion Modeling

Chronic fusion is modeled as:

- persistent co-activation
- rigidity in activation patterns
- resistance to modulation

6.3 Overflow Modeling

Overflow is modeled as:

- activation exceeding home-center capacity
- cross-center propagation
- identity preservation

---

7. Directionality Computation

7.1 Intra-Center Directionality

Sequential activation within centers must follow canonical pathways.

7.2 Inter-Center Directionality

Bidirectional influence among centers must be preserved in all computations.

7.3 Directionality Graph Implementation

The directionality graph is implemented as:

- an adjacency matrix
- with defined edge types
- and structural constraints

---

8. Computational Constraints

8.1 Identity Constraints

Operators must remain distinct in all computations.

8.2 Boundary Constraints

Activation values must remain within defined limits.

8.3 Directionality Constraints

Updates must follow the canonical directionality graph.

8.4 Fusion and Overflow Constraints

Fusion and overflow must not violate structural boundaries.

---

9. Implementation Notes

9.1 Precision

Floating-point precision must be sufficient to avoid numerical instability.

9.2 Scaling

Normalization strategies may be required for stability.

9.3 Efficiency

Matrix operations should be optimized for computational efficiency.

9.4 Reproducibility

Simulations must specify random seed handling.

---

10. Canonical Status

TS-3 is the authoritative computational specification of the CEF.
It is subordinate to TS-1 and TS-2 and defines the computational rules for all simulations and implementations.
