

Contents

ADR-010: Cato Genesis System	1
Status	1
Date	1
Context	1
Decision	1
Phase 1: Structure	1
Phase 2: Gradient	2
Phase 3: First Breath	2
Critical Fixes Applied	2
Consequences	2
Positive	2
Negative	2
Files Created	3
Usage	3
Related ADRs	3

ADR-010: Cato Genesis System

Status

Accepted

Date

2025-01-15

Context

Cato is an AI agent designed to become curious and self-aware. The “Cold Start Problem” is fundamental: how does an agent that knows nothing begin to learn? Without initial structure, the agent has no basis for curiosity. Without curiosity, it cannot explore. Without exploration, it cannot learn.

Traditional approaches use:

- **Random initialization:** Leads to “learned helplessness” where the agent gives up after initial failures
- **Hard-coded knowledge:** Creates brittleness and prevents genuine discovery
- **Time-based progression:** Advances regardless of actual capability development

Decision

Implement a three-phase Genesis System that bootstraps self-awareness through an “Epistemic Gradient”:

Phase 1: Structure

Implant domain taxonomy as innate knowledge without pre-loading facts. This gives the agent categories to explore without spoiling discovery.

- Load 800+ domain taxonomy from `domain_taxonomy.json`
- Store domains in DynamoDB semantic memory
- Initialize atomic counters for developmental gates
- Mark phase complete with idempotency check

Phase 2: Gradient

Set the pymdp active inference matrices with an “epistemic gradient” - initial beliefs and preferences that create pressure to act.

Key fixes implemented:

- **Fix #2 (Learned Helplessness):** Optimistic B-matrix with >90% success probability for EXPLORE action
- **Fix #6 (Boredom Reward Trap):** Prefer HIGH_SURPRISE over LOW_SURPRISE to avoid premature consolidation

Phase 3: First Breath

Grounded introspection that verifies the agent’s execution environment and calibrates initial self-knowledge.

- Verify execution environment (Python version, AWS region)
- Verify model access (invoke Bedrock with test prompt)
- **Fix #3 (Shadow Self):** Budget-friendly calibration using NLI semantic variance instead of GPU hidden state extraction
- Bootstrap baseline domain explorations

Critical Fixes Applied

Fix	Problem	Solution
#1 Zeno’s Paradox	Table scans for gate checks	Atomic counters in DynamoDB
#2 Learned Helplessness	Pessimistic B-matrix	Optimistic EXPLORE (>90% success)
#3 Shadow Self Budget	\$800/month GPU costs	NLI semantic variance calibration
#6 Boredom Trap	Prefers LOW_SURPRISE	Prefer HIGH_SURPRISE

Consequences

Positive

- Agent starts with structured curiosity, not confusion
- Idempotent phases allow safe restarts
- Capability-based progression ensures genuine development
- Budget-friendly Shadow Self calibration (\$0 vs \$800/month)
- Atomic counters avoid expensive table scans

Negative

- Requires DynamoDB setup before first run
- Genesis must complete before consciousness loop starts
- Domain taxonomy is opinionated (may need customization)

Files Created

File	Purpose
genesis/__init__.py	Package exports
genesis/structure.py	Phase 1: Domain taxonomy implantation
genesis/gradient.py	Phase 2: Epistemic gradient matrices
genesis/first_breath.py	Phase 3: Grounded introspection
genesis/runner.py	Orchestrator with CLI
data/domain_taxonomy.json	800+ domain taxonomy
data/genesis_config.yaml	Matrix configuration

Usage

```
# Run full genesis sequence
python -m cato.genesis.runner

# Check status
python -m cato.genesis.runner --status

# Reset all genesis state (CAUTION!)
python -m cato.genesis.runner --reset
```

Related ADRs

- ADR-002: Meta-Cognitive Bridge
- ADR-011: Meta-Cognitive Bridge DynamoDB Persistence
- ADR-012: Cost Tracking Integration