Item #245 - DBM Structure Intelligence Agent - From Algorithms to Agents, from Intelligence to Transferable Structure

Conversation: DBM Structure Intelligence Agent

20260122

Authors: Sizhe Tan & GPT-Obot

DBM-COT ITEM #245 DBM Structure Intelligence Agent

From Algorithms to Agents, from Intelligence to Transferable Structure

Abstract

As artificial intelligence systems increasingly replace low-level programming labor, the primary bottleneck of intelligence evolution is shifting from *algorithm implementation* to *structural decision-making in real applications*. This document introduces the concept of the DBM Structure Intelligence Agent (SIA) — a new class of intelligent agent designed not merely to *produce answers*, but to *construct, preserve, and transfer structural intelligence*.

We argue that traditional AI service paradigms (APIs, models, tools, or even agents) are insufficient for long-term intelligence evolution. Instead, DBM proposes a higher service form:

An agent that solves problems *and* delivers the underlying structural program so that intelligence can be inherited, modified, and evolved by others.

This item is written for readers with no prior DBM background.

1. The 2026 DBM Roadmap in One Sentence

DBM's long-term roadmap consists of three layers:

- DBM-COT (Chain of Thought) — principles, constraints, and structural intelligence theory
- DBM-COA (Chain of Algorithms) — concrete algorithms and runtime mechanisms
- DBM-COS (Chain of Solutions) — real-world applications and systems

The DBM Structure Intelligence Agent is the *core carrier* of DBM-COA into DBM-COS.

2. Why Algorithms Alone Are No Longer Enough

Open-sourcing algorithms is necessary, but no longer sufficient.

2.1 Strengths of Algorithm-Centric Transmission

- Preserves deep technical knowledge
- Enables long-term academic continuity

2.2 Structural Limitations

- Extremely high entry barrier
- Very slow evolutionary feedback
- Unpredictable adoption and impact
  Increasingly few people interact directly with algorithms

As AI systems automate low-level coding, fewer humans participate at the algorithm layer, while more humans interact at the application and decision layer.

3. Intelligence Has Moved from Code to Structure

Modern intelligence challenges are no longer: “How
do we implement this algorithm?” They are:

-
- How do we choose *what structure to search*?
- When should we stop?
- Which trade-off is acceptable?
- Which path is feasible but not optimal?
- How do we explain decisions to others?

These are structural intelligence problems, not coding problems.

4. Why an Agent — Not Just a Tool or Model

An Agent introduces properties that algorithms alone cannot provide:

- Persistence over time
- Interaction with incomplete information
- Path dependence
- Budget awareness
- Decision stopping rules
- Structural explanation

DBM algorithms are natively agent-friendly:

- Differential Trees
- Two-Phase Search
- Stop Rules
- Feasible Path reasoning
- CCC (Common Concept Core) voting

DBM was structurally aligned with agents *before agents became mainstream*.

5. Beyond "Agent as a Service": A Higher Paradigm

Most current AI systems follow one of three paradigms:

1. Result-as-a-Service — return answers
2. Tool-as-a-Service — expose functions
3. Agent-as-a-Service — run autonomous workflows

All three share a fatal limitation:

The intelligence disappears when the service stops.

6. DBM's Proposal: Structural Intelligence Transfer

DBM introduces a higher service form: An agent that not only solves the problem, but delivers the *structural program* it constructed.

This includes:

- A runnable program skeleton
- Structural rationale
- Evidence and decision paths
- Stop conditions and constraints
- Explicit extension points

This allows:

- Multiple teams to continue independently
- Intelligence inheritance without re-training
  Divergent evolution from a shared structure

In short:

the agent passes the torch, not just the result.

•

7. Why DBM Is Uniquely Capable of This

DBM differs from mainstream AI in three fundamental ways:

7.1 Structure Is Explicit

DBM structures are not hidden in weights:

- Trees
- Paths
- Evidence
- Concepts

They can be exported, inspected, and reused.

7.2 Minimal Evolution Threshold

DBM explicitly controls complexity:

- No over-engineering
- No premature optimality • Human-handoff is a design constraint

7.3 Feasible Beats Optimal

DBM optimizes for:

- Interpretability
- Transferability
- Continuation

Not one-shot optimal answers.

8. The DBM Structure Intelligence Agent Stack L0 — Structural Substrate (Invisible)

The agent is organized into three layers:

- Differential Trees
- Metric routing

- CCC / IR / Evidence
- Stop rules and pruning

L1 — Structural Reasoning Core

- Problem → Structure mapping
- Two-Phase Search orchestration
- Feasible Path generation
- Structural consistency checking
- Explanation assembly

L2 — Interaction & Integration

- APIs / CLI / IDE
- Local or cloud deployment
- Optional LLM collaboration
- Application adapters

9. The Ultimate Goal

The highest form of DBM service is not automation.

It is structural inheritance.

DBM does not aim to replace human intelligence, but to ensure that intelligence can survive, transfer, and evolve across people, teams, and generations.

10. Positioning Statement

LLM agents perform tasks.

-

DBM Structure Intelligence Agents create successors.

DBM-COT 条目 #245（中文版） DBM 结构智能 Agent 从算法到 *Agent*，从智能到可传承结构

摘要

随着 AI 逐步接管底层编程劳动，智能演化的核心瓶颈，正在从“算法实现”转移到“现实世界中的结构性决策”。

本文提出 DBM Structure Intelligence Agent（结构智能 Agent） 的概念：

它不是为了给出答案，而是为了构建、保存并传递结构智能。

DBM 认为，传统的 AI 服务形态——无论是模型、API、工具还是 Agent——都不足以支撑长期智能演化。

因此，DBM 提出一种更高层级的服务范式：

Agent 在解决问题的同时，把形成的结构程序交付给应用者，使智能能够被继承、修改和继续演化。

本文面向没有任何 DBM 背景的新读者。

1. DBM 的三阶段路线图

- DBM-COT：思想、原则、结构智能理论
- DBM-COA：算法、机制、运行时系统
- DBM-COS：应用与解决方案

结构智能 Agent 是 DBM-COA 进入真实世界的关键载体。

2. 为什么仅靠算法开源已经不够

算法传承有价值，但存在根本局限：

- 参与门槛极高
- 演化周期极长

- 反馈不可控
- 越来越少的人直接接触算法与此同时，应用端与决策端的参与者正在爆炸式增长。

3. 智能的重心已经从“代码”转向“结构”

今天真正困难的问题不是“怎么写代码”，而是：

- 搜索什么结构
- 什么时候停
- 如何取舍
- 如何解释
- 如何交接这是结构智能问题。

4. 为什么必须是 Agent

Agent 引入了算法无法单独承担的能力：

- 时间连续性
- 不完全信息
- 路径依赖
- 成本与风险控制
- 停止规则
- 解释能力

而 DBM 的核心算法，天生适合 Agent。

5. 超越“Agent 即服务”

现有 AI 服务无论多智能，都有一个共同缺陷：服务结束，智能消失。

DBM 要解决的是：智能如何被接力。

6. DBM 的 高服务形态：结构智能迁移

DBM Agent 的 高形态是：

解决问题 + 输出可运行、可解释、可接管的结构程序。

这样，智能不再被“消费”，而是被继承与分叉演化。

7. 为什么只有 DBM 能系统性做到这一点

- DBM 的结构是显式对象

- DBM 以 小进化门槛为约束
- DBM 追求“可继续”，而非“一次 优”

8. 三层 Agent 技术栈

- L0：不可见的结构基底

- L1：结构推理核心 • L2：对外接口与集成

9. 终极目标

DBM 不是为了自动化世界，而是为

了让智能可以被传承。 10. 定位

总结

LLM Agent 替你做事，

DBM 结构智能 Agent 帮你留下能继续做事的结构。