ITEM #219 - **Time-Series IR Top Requirements - Task–Action Dual ACLM and Flow-Shop Runtime Architecture**

Conversation：Time-Series IR Requirements

20260109

Authors: Sizhe Tan & GPT-Obot

DBM-COT ITEM #219

Time-Series IR Top Requirements

Task–Action Dual ACLM and Flow-Shop Runtime Architecture

Author: Digital Brain Model (DBM)
Category: Time-Series IR · ACLM · Structural Intelligence
Level: L1 (Architectural & Methodological)

Abstract (EN)

This ITEM formalizes the **top-level requirements** for a next-generation Time-Series IR (Intermediate Representation) system, grounded in DBM structural intelligence and ACLM Task–Action duality.

The objective is to elevate Time-Series IR from a fixed, pattern-centric event generator into a **configurable, evolvable research factory**, aligned with senior quantitative and discretionary analysts' workflows and capable of entering large-scale APTGOE-style structural search and evolution.

This document defines the required architectural principles, runtime semantics, governance constraints, and industrial-engineering analogies necessary to ensure long-term competitiveness, extensibility, and analytical credibility.

---

1. Background and Motivation

Recent DBM Time-Series IR work has completed the **event generation foundation layer** (“lower body”):
a unified, pluggable extractor architecture capable of hosting dozens of event-generation patterns under a common runtime.

The remaining challenge is the **upper structure** (“upper body”):
how Time-Series IR should be **used, composed, governed, and evolved** in real analytical environments.

Without explicit top-level requirements, Time-Series IR risks degenerating into:

- rigid “fixed 套餐” pipelines,
- brittle re-programming cycles,
- or short-lived, pattern-specific tools.

This ITEM addresses that gap.

---

2. Target Users and Application Scope

The primary application 主体 is **senior market analysts**, including:

- Experience-driven discretionary analysts,
- Quantitative and systematic researchers,
- Hybrid users combining intuition, structure, and computation.

Accordingly, Time-Series IR must function as a **research workbench**, not a static model package.

Key implications:

- Fixed pipelines are unacceptable.
- Rapid hypothesis iteration is mandatory.
- Structural transparency and reproducibility are non-negotiable.
- The system must remain competitive against future algorithms, not merely current ones.

---

3. Core Design Constraint: Variability and Evolution

Financial time-series systems are inherently:

- non-stationary,

- regime-dependent,
- adversarial,
- and structurally evolving.

Therefore:

3.1 No Fixed 套餐 Principle

Time-Series IR must not hard-code:

- extractor sets,
- execution order,
- or pattern semantics.

3.2 Rapid Structural Mutation

Iteration must **not require recompilation** or deep code modification.

The only viable path is:

- **JSON-driven configuration**, and
- **runtime pipeline self-composition**.

---

4. Task–Action Dual ACLM Architecture

Time-Series IR is formally defined as a **dual-layer ACLM system**:

4.1 Task Layer ACLM (Configuration / Intent)

- Expressed as JSON "recipes".
- Specifies:
  - input curves,
  - extractor selection,
  - execution graph or sequence,
  - policies and constraints.

This layer represents **what is being asked**.

4.2 Action Layer ACLM (Runtime / Execution)

- The Time-Series IR pipeline runtime.
- Dynamically composes:
  - extractors,
  - buses,
  - event flows,

- guards and caches.

This layer represents **how it is executed**.

Separation is mandatory:
Task evolution must never compromise Action-layer correctness.

---

5. Canonical Usage Form

A generalized usage form is defined as:

```
Event[t, j] = TimeSeriesIR(
    sourceCurves,
    patternExtractors,
    extractorCallingGraphOrSequence
)
```

In practice, this expands into a four-stage lifecycle:

1. **Recipe (JSON)**
2. **Plan (compiled pipeline graph + resolved parameters)**
3. **Run (runtime execution with guards and caches)**
4. **Artifacts (events, evidence, summaries, run manifest)**

Artifacts, not raw events, are the true analytical output.

---

6. Industrial Engineering Analogy: Flow Shop / Job Shop

From an Industrial Engineering / Operations Research perspective,
Time-Series IR is a **research factory**.

6.1 Factory Components Mapping

Industrial Concept	Time-Series IR
Workpiece	Curve / CurveSlice
Machine	Extractor / Aggregator
Fixture	Windowing, Alignment, Normalization
Conveyor / Bus	DataBus / EventBus
Routing	Calling Graph / Sequence
QC	Guards, Assertions, Sanity Checks
Process Sheet	JSON Recipe

6.2 Dual-Bus Principle

- **DataBus**: curves, windows, features, observations
- **EventBus**: events, revisions, dependencies

"Events generating events" is treated as a **first-class flow**, not an exception.

---

7. Top Requirements Summary (R1–R8)

R1. Composition-First Pipelines

Pipelines must be graph-composable, not code-fixed.

R2. Event→Event Dependency Graphs

Ordering is semantic, not arbitrary.

R3. Reproducibility by Construction

Every run must emit a complete run manifest.

R4. Multi-Layer IR

Observation → Event → Summary / Wrapper.

R5. Built-in Explanation Chains

Every event must justify itself structurally.

R6. Policy Externalization

Extractors measure; policies decide.

R7. APTGOE-Scale Performance

Batching, caching, parallelism, and safety guards are mandatory.

R8. Governance and Safety Rails

Versioning, auditing, quotas, and failure containment.

---

8. Strategic Significance for APTGOE

As Time-Series IR enters APTGOE-style evolution:

- Search space explodes (recipes × assets × horizons).
- Structural governance becomes existential.
- Task–Action separation ensures **controlled evolution**, not chaos.

This architecture allows:

- aggressive structural search,
- without sacrificing correctness,
- interpretability,
- or institutional credibility.

---

9. Conclusion

Time-Series IR is no longer a pattern extractor.

It is a **structural intelligence substrate**—
a configurable, governed, evolvable analytical factory.

ITEM #219 defines the **non-negotiable architectural contract** required to support that future.

DBM-COT ITEM #219（中文版）

时间序列 IR 顶层需求

Task–Action 双层 ACLM 与"流水线车间"运行时架构

摘要

本文系统性地提出了 **下一代 Time-Series IR（中间表示）系统的顶层需求**，基于 DBM 的结构智能理念与 ACLM 的 Task–Action 双层架构。

目标是将 Time-Series IR 从"固定事件生成器"，提升为 **可配置、可演化、可治理的研究工厂**，对标华尔街资深分析师的真实工作流，并能够进入 APTGOE 式的大规模结构搜索与进化体系。

---

1. 背景与问题界定

当前 DBM 的 Time-Series IR 已完成最复杂的**事件生成基础层（下半身）**：

- 统一的 extractor 接口，
- 可插拔的事件生成机器，
- 稳定的 runtime 骨架。

真正决定系统上限的，是尚未完全固化的 **“上半身”**：

- 如何使用，
- 如何组合，
- 如何治理，
- 如何演化。

本 ITEM 正是对此进行结构性封顶。

---

2. 应用主体的现实约束

Time-Series IR 的目标用户是：

- 经验型资深分析师，
- 数字型量化研究者，
- 以及两者融合的高级用户。

他们需要的是 **研究工艺系统**，而不是模型套餐。

---

3. 不可回避的核心事实：系统必然多变

金融时间序列：

- 非平稳，

- 强对抗，
- 结构不断迁移。

因此：

- 固定套餐是死路；
- 重新编程迭代是不可接受的成本。

---

4. Task–Action 双层 ACLM 定义

Task 层（JSON 配方）

描述“要做什么”：

- 数据来源，
- extractor 选择，
- 执行图，
- 策略与约束。

Action 层（IR Runtime）

描述“如何做”：

- 自动组装流水线，
- 管理事件流，
- 执行与防护。

两层必须严格解耦。

---

5. 使用范式的正式升级

Time-Series IR 的真实生命周期是：

Recipe → Plan → Run → Artifacts

而不是只返回一个 Event 表。

---

6. 流水线车间模型（工业工程视角）

Time-Series IR = 研究型 Flow Shop / Job Shop。

- 曲线是工件，
- extractor 是机器，
- JSON 是工艺卡，
- DataBus / EventBus 是输送系统。

事件生成事件，是常态而非特例。

---

7. 顶层需求清单（R1–R8）

- **R1**：组合优先
- **R2**：事件依赖图
- **R3**：可复现性
- **R4**：多层 IR
- **R5**：内建解释链
- **R6**：策略外置
- **R7**：APTGOE 级扩展
- **R8**：治理与安全护栏

---

8. 对 APTGOE 的战略意义

Task–Action 双层 ACLM 的真正价值在于：

允许结构疯狂进化，但把执行与治理牢牢锁死。

这正是可控智能进化的前提。

---

9. 总结

Time-Series IR 不再是工具。

它是 **结构智能的研究工厂内核**。

ITEM #219 为其未来演化，设定了不可动摇的上限边界。
