

BPMN-Aware Large-Scale Process Analysis through LLM Tool Orchestration

Abstract

This paper proposes a novel approach for analyzing large volumes of business process data by combining three concepts: (1) pre-generated process summaries ("Project Insights"), (2) BPMN process definitions as discoverable metadata, and (3) LLM-driven tool orchestration inspired by Recursive Language Models. The approach enables natural language queries across thousands of workflow instances without hitting context window limitations, while leveraging existing process knowledge encoded in BPMN models.

1. Problem Statement

Enterprise workflow systems accumulate vast amounts of process data. A typical scenario:

- 10,000+ completed workflow instances over 6 months
- Multiple workflow types (orders, complaints, quotes, support tickets)
- Rich contextual data per instance (documents, communications, decisions)

Challenge: Users want to ask analytical questions like *"How did our sales perform in 2025?"* — questions that implicitly span multiple workflow types and require correlation of thousands of instances.

Limitations of current approaches:

Approach	Limitation
Direct LLM query	Context window exceeded
Pre-built reports	Cannot handle ad-hoc questions
Manual analysis	Time-consuming, not scalable
Simple RAG	Retrieves fragments, misses cross-process patterns

2. Proposed Solution

2.1 Core Idea

Treat the LLM not as a processor of raw data, but as an **orchestrator** that:

1. **Discovers** available workflow types through BPMN metadata
2. **Decides** which processes are relevant to a given question
3. **Retrieves** condensed summaries in manageable chunks
4. **Correlates** across workflow types when needed
5. **Synthesizes** findings into coherent answers

2.2 Key Enablers

A) Project Insights (Pre-Condensed Summaries)

Each workflow instance already generates a structured summary at key process steps:

```
Project Insight:
Event: A quotation request was received from Tom Brack of TechBridge,
Tokyo for electronic equipment suitable for Arctic conditions...
Unusual/Important Note: Sales team suggested offering products still
in development...
```

This reduces token volume by ~95% while preserving analytical value.

B) BPMN as Discoverable Metadata

Process definitions provide machine-readable context:

- Available workflow types and their relationships
- Process steps and decision points
- Data fields captured at each stage
- Semantic documentation

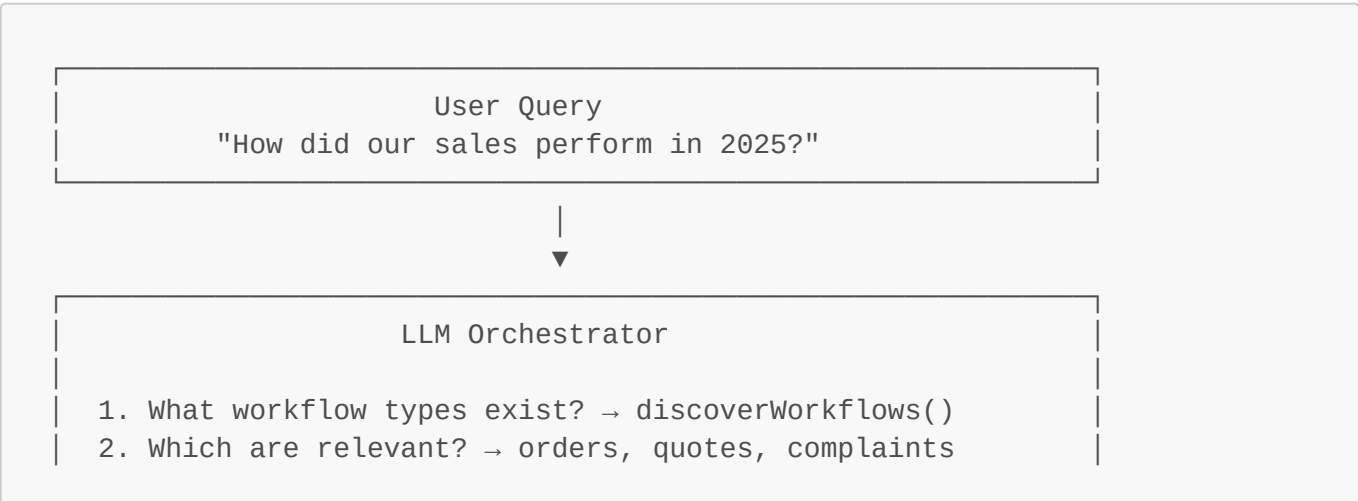
The LLM can query this metadata to understand *what exists* before deciding *what to analyze*.

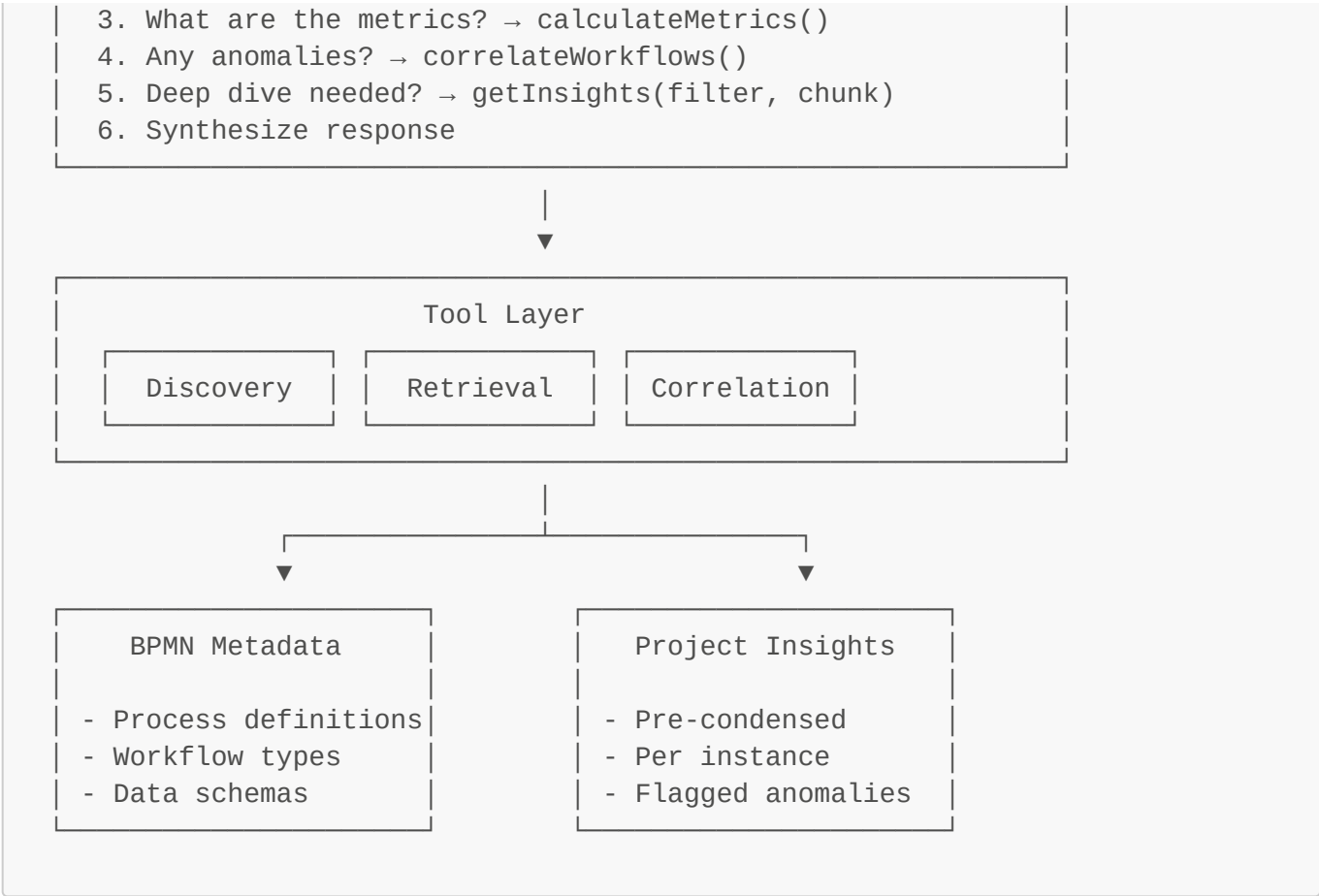
C) Tool-Based Access Pattern

Instead of feeding data directly into the context window, provide callable tools:

Tool Category	Purpose
Discovery	List workflow types, describe process structure
Retrieval	Fetch insights by type, filter, pagination
Correlation	Find relationships across workflow types
Aggregation	Calculate cross-process metrics

3. Architecture Overview





4. Relation to Existing Research

4.1 Recursive Language Models (Zhang et al., 2025)

RLM proposes treating long contexts as external variables accessible through a REPL environment. Our approach adopts the core insight — **data as environment, not input** — but replaces arbitrary code execution with controlled tool interfaces.

RLM (Paper)	Our Approach
Python REPL	Defined tool interface
Arbitrary code	Controlled operations
Generic text context	Structured BPMN + Insights
LLM decides chunking	Tools handle pagination

4.2 Function Calling / Tool Use

Standard LLM tool use focuses on single-purpose functions. Our contribution is the **discovery layer**: the LLM first learns what workflow types exist, then decides which to query — enabling truly ad-hoc analysis without pre-built report logic.

5. Required Tool Categories

5.1 Discovery Tools

Tool	Input	Output
discoverWorkflows	period	List of workflow types with counts
describeWorkflow	workflowType	Process steps, data fields, sample

5.2 Retrieval Tools

Tool	Input	Output
getInsights	type, filter, chunk	Paginated insight summaries
getStatistics	type, period	Aggregate counts and metrics

5.3 Correlation Tools

Tool	Input	Output
correlateWorkflows	typeA, typeB, key	Cross-process relationships
calculateMetrics	types[], period	KPIs spanning multiple types

5.4 Session Tools

Tool	Input	Output
storeAnalysis	category, content	Persist intermediate findings
getStoredAnalyses	—	Retrieve for final synthesis

6. Expected Benefits

1. **Unlimited Scale:** No context window constraints; 10K or 100K instances handled identically
2. **Ad-Hoc Flexibility:** New question types require no new code — LLM discovers relevant workflows
3. **Cross-Process Intelligence:** Automatic correlation (e.g., complaints ↔ orders) reveals hidden patterns
4. **Controlled Execution:** Unlike open REPL, tool interface ensures security and auditability
5. **Incremental Enhancement:** New workflow types automatically become queryable

7. Open Questions

- **Chunking Strategy:** Fixed size vs. semantic boundaries vs. LLM-determined?
- **Caching:** How to cache intermediate analyses across similar queries?
- **Confidence:** How does the LLM signal when data is insufficient?
- **Cost Control:** Maximum tool calls per query? Budget-aware orchestration?

8. Next Steps

1. Define minimal viable tool interface (Discovery + Retrieval)
 2. Prototype with single workflow type
 3. Evaluate chunking strategies empirically
 4. Add correlation tools
 5. Benchmark against direct-context baseline
-

References

- Zhang, A.L., Kraska, T., Khattab, O. (2025). *Recursive Language Models*. arXiv:2512.24601
- OpenAI (2024). *Function Calling Documentation*.
- OMG (2013). *Business Process Model and Notation (BPMN) Version 2.0.2*