

УДК 004.8:004.89

DOI: <https://doi.org/10.17721/AIT.2025.2.03>

Danylo DVOICHENKOV, PhD Student

ORCID ID: 0009-0007-1935-6743

e-mail: kmwp@irtc.org.ua

Institute of Information Technologies and Systems of the National Academy of Sciences of Ukraine, Kyiv, Ukraine

Oleksandr MARCHENKO, DSc (Phys. & Math.), Prof.

ORCID ID: 0000-0002-5408-5279

e-mail: omarchenko@univ.kiev.ua

Institute of Information Technologies and Systems of the National Academy of Sciences of Ukraine, Kyiv, Ukraine

Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

TO FERRET OUT A GLOBAL METASCAFFOLDING STANDARD: METASCAFFOLD TAXOFRAME TEMPLATE CASE AS A PARADIGMATIC DEMONSTRATION OF THE APPROACH

Background. The proliferation of Large Language Model (LLM)-based AI systems has led to widespread adoption of scaffolded reasoning frameworks that enhance model capabilities through structured methodologies including chained prompts, multi-agent coordination, and external tool integration. However, the ad-hoc nature of current scaffolding design processes creates significant challenges in interoperability, scalability, and reproducibility across diverse AI workflows. This paper introduces the MetaScaffold paradigm – a higher-level abstraction for structuring and orchestrating AI reasoning processes that provides reusable, systematic, domain-agnostic principles for organizing knowledge and reasoning steps. We present the MetaScaffold TaxoFrame Template (MSTFT) as a practical instantiation of this paradigm, specifically designed for taxonomic processing tasks, demonstrating how standardized meta-structures can improve AI system modularity and effectiveness while maintaining flexibility across different domains and applications.

Methods. We formalize MetaScaffolds as structured orchestration mechanisms that dynamically integrate meta-level prompting, agent-based reasoning, self-reflective loops, and external knowledge retrieval through a central decision engine. The MSTFT framework implements this paradigm with core components including: (1) Meta-prompts and Recursive Meta-prompts for iterative refinement cycles, (2) modular agent systems for specialized reasoning capabilities, (3) memory and context management for semantic consistency, and (4) configurable model parameters for task optimization. Our implementation follows a structured procedure involving preprocessing and retrieval, meta-prompt assembly, dynamic model selection, and iterative reasoning with validation. We evaluated MSTFT against Single-Pass LLM (SP-LLM) and Iterative Prompting (IP) baselines across multiple domains (Software, Wood, Biology, Disease) using reference taxonomies constructed from WordNet hypernyms and ConceptNet derived terms, with evaluation focused on conceptual relevance, hierarchical coherence, instruction adherence, and generation efficiency.

Results. Experimental validation demonstrates MSTFT's superior performance across all evaluation metrics. In the Software domain, MSTFT generated 80-90 well-structured and semantically distinct subconcepts compared to SP-LLM (~30 concepts) and IP (~50 concepts), which exhibited issues with redundancy and ambiguity. MSTFT consistently maintained three-level hierarchical depth as specified, while baseline methods produced flat or inconsistent structures. The framework achieved optimal balance between quality and cost efficiency (~\$0.03 per taxonomy) compared to SP-LLM (~\$0.01) and IP (~\$0.10). Cross-domain evaluation confirmed MSTFT's generalizability, producing more differentiated and structured hierarchies across Wood, Biology, and Disease domains. Qualitative analysis revealed MSTFT's outputs demonstrated superior conceptual relevance, hierarchical coherence, and minimal logical errors, with approximately 25-30 valid subconcepts per level across three hierarchical levels.

Conclusions. This work introduces the MetaScaffold paradigm and demonstrates its practical implementation through MSTFT, addressing critical gaps in existing AI scaffolding methodologies. MSTFT successfully integrates meta-level reasoning, iterative refinement, and multi-agent orchestration into a unified, interoperable framework that significantly improves taxonomy construction quality while maintaining computational efficiency. The framework's modular design, semantic validation mechanisms, and standardized approach provide a foundation for broader applications in AI reasoning tasks including ontology alignment, long-term planning, and multi-step question answering. Future research directions include enhanced semantic validation, multi-modal reasoning support, and expanded empirical evaluation across diverse domains and languages. While limitations exist regarding LLM dependencies and computational overhead, MSTFT establishes a promising paradigm for standardized, reusable AI scaffolding frameworks that balance flexibility with systematic structure.

Keywords: MetaScaffold, taxonomy construction, meta-prompting, multi-agent systems, large language models, AI reasoning frameworks, knowledge organization, iterative refinement, ontology engineering.

Background

Introduction and Motivation

The rapid expansion of LLM-based AI architectures has resulted in widespread adoption of scaffolded reasoning frameworks-structured methodologies that augment raw model outputs through techniques such as chained prompts, multi-agent coordination, and external tool integration (e.g., via reasoning-and-acting paradigms (Yao et al., 2023)). While such scaffolds significantly improve model capabilities, their growing complexity and the ad-hoc nature of their design processes have created substantial challenges related to interoperability, scalability, and reproducibility across different AI-driven workflows.

Current scaffolding methods vary from simple prompt chaining strategies-such as Chain-of-Thought reasoning (Gao et al., 2024; Zhang, Yuan, & Yao, 2024)-to complex multi-agent architectures involving retrieval-augmented generation (RAG), agent orchestration, and structured knowledge representations (Liu et al., 2024; Ma et al., 2024). Despite demonstrated effectiveness,

a standardized framework for scaffold design remains elusive, making it difficult to generalize successful techniques or ensure seamless integration across diverse AI applications.

To address these critical gaps, we introduce the **MetaScaffold** paradigm—a higher-level abstraction for structuring and orchestrating AI reasoning processes. MetaScaffolds generalize traditional task-specific scaffolding frameworks by providing reusable, systematic, domain-agnostic principles for organizing knowledge and reasoning steps. This approach allows AI systems to leverage standardized meta-structures while avoiding rigid constraints typical of conventional hierarchical structures.

This paper makes the following key contributions:

1. Formalization of MetaScaffolds as a generalized methodology for AI-driven reasoning and agent orchestration.
2. A clear taxonomy and specification schema distinguishing traditional scaffolds from MetaScaffolds.
3. Integration of Meta-Prompting and Recursive Meta-Prompting for dynamic, self-improving AI reasoning.
4. Design of an LLM-agnostic MetaScaffold framework ensuring modularity, flexibility, and semantic consistency.
5. Experimental validation through a practical implementation, demonstrating effectiveness in taxonomy-related tasks and retrieval-augmented generation (RAG) applications.

The remainder of the paper is organized as follows: Section 2 reviews related literature on scaffolds, meta-prompting, and multi-agent orchestration. Section 3 defines the MetaScaffold paradigm and proposes a formal standardized schema. Section 4 presents MSTFT, detailing its design principles, implementation strategies, and capabilities. Section 5 describes empirical experiments validating the MetaScaffold approach in taxonomy construction tasks. Section 6 concludes with insights and future research directions.

Related Works.

Meta-Prompting and Meta-Reasoning.

Recent developments in prompting methodologies have introduced **meta-prompting** – structured meta-level prompts that guide AI models' reasoning processes beyond simple few-shot or chain-of-thought methods. Zhang et al. (Zhang, Yuan, & Yao, 2024) presented a syntax-oriented meta-prompting strategy grounded in type and category theory, enhancing model reasoning by decomposing complex tasks into reusable meta-structures. Similarly, Suzgun and Kalai (Suzgun & Kalai, 2024) extended meta-prompting by orchestrating multi-agent interactions within a single model, enabling it to manage specialized internal “expert” prompts dynamically; this resulted in marked performance improvements (Gao et al., 2024).

Complementary to meta-prompting, **meta-reasoning prompting (MRP)** approaches (Gao et al., 2024; Bhutani et al., 2024) employ a metacognitive strategy, empowering LLMs to dynamically select appropriate reasoning techniques based on task complexity. This mirrors human metacognitive abilities, allowing the model to decide whether to, for example, break a problem into steps or consult external information.

Retrieval-Augmented Generation and Self-Reflective Strategies

Beyond prompt formulation, recent research emphasizes self-reflection and external knowledge integration. Self-reflective cycles—such as the Environmental Feedback-based Self-Correction framework (ERASER) proposed by Huang et al. (Huang et al., 2024)—iteratively refine model outputs through explicit evaluation, reflection, and revision steps. These cyclic processes (often summarized as *Solve* → *Evaluate* → *Reflect* → *Revise*) enable models to identify and correct errors or hallucinations in their responses. Furthermore, retrieval-augmented generation techniques have been augmented with metacognitive modules: for instance, Meta-cognitive RAG (MetaRAG) (Zhou et al., 2024) allows a model to plan and control its use of external knowledge sources dynamically, deciding what to retrieve and when, based on an ongoing assessment of the task at hand.

Multi-Agent Orchestration and Design Patterns

Another avenue of related work integrates multiple LLM-based agents in a coordinated architecture. Recent surveys (Liu et al., 2024; Ma et al., 2024) highlight the importance of modular multi-agent systems for decomposing complex tasks, wherein each agent specializes in a distinct role. Formal coordination patterns, such as behavior trees introduced by Kelley et al. (Kelley, 2024), provide structured blueprints to orchestrate interactions among agents via defined sequences, fallback strategies, and decision nodes—significantly reducing brittleness and improving reliability.

Collectively, these advancements underline the need for a higher-level, standardized framework capable of systematically integrating meta-prompting, meta-reasoning, and multi-agent orchestration techniques. Our proposed **MetaScaffold** methodology addresses this need, offering an abstract, reusable framework that unifies these approaches into a cohesive standard—enabling enhanced modularity, interoperability, and generalizability across diverse AI applications.

Methods

Formalizing the MetaScaffold Paradigm

The **MetaScaffold** paradigm generalizes existing scaffolding approaches by abstracting task-specific reasoning processes into reusable, domain-agnostic *meta-structures*. Formally, we define a MetaScaffold as a structured orchestration mechanism that dynamically integrates meta-level prompting, agent-based reasoning, self-reflective loops, and external knowledge retrieval, with the goal of standardizing interactions among foundational language models (LLMs), external tools, and contextual data.

At the core of a MetaScaffold lies a *decision engine*—a central orchestrator responsible for dynamically assembling reasoning components based on task requirements and environmental context (see Fig.1). This decision engine interfaces with multiple resources, including:

1. *Meta-prompts and Recursive Meta-prompts*: Structured instructions that guide the model's reasoning through iterative refinement cycles, enhancing adaptive problem-solving capabilities by embedding self-review and adjustment steps.
2. *Agent Modules and External Tools*: Modular components enabling specialized reasoning capabilities or providing access to computational tools and external data sources (e.g., knowledge bases, search engines).
3. *Memory and Context Management*: Persistent storage and retrieval mechanisms that maintain semantic consistency and track historical interactions or intermediate results throughout the reasoning process.
4. *Model Parameters and External Configurations*: Adjustable parameters and external configurations that optimize model behavior according to specific task domains and contexts (for example, switching to a domain-specific LLM for technical queries).

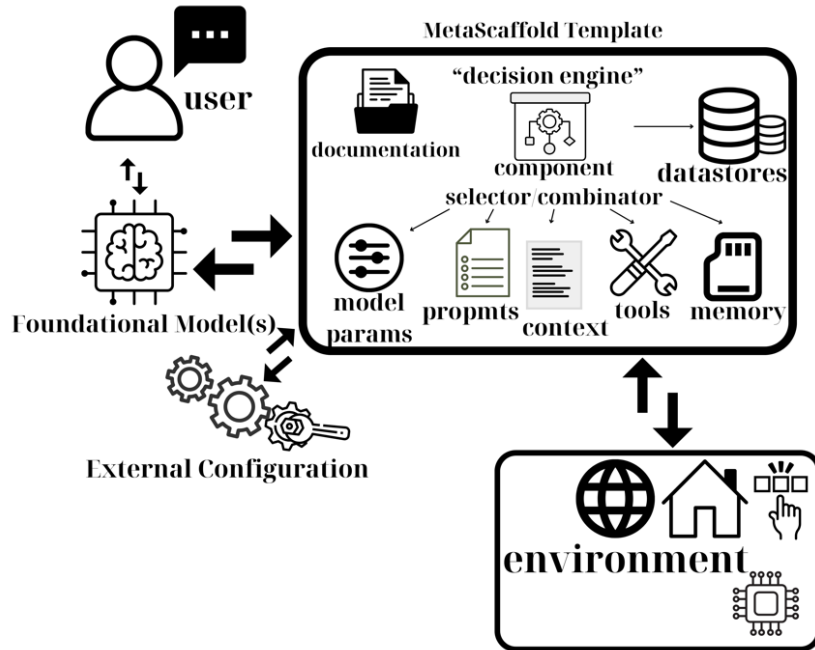


Figure 1 Generalized MetaScaffold Template architecture

MetaScaffolds explicitly incorporate **meta-reasoning**, enabling an LLM to dynamically choose and execute reasoning strategies best suited to a given task. Following Gao et al. (Gao et al., 2024), this meta-reasoning process can be viewed in two phases: first, the model reflects on the problem and selects an appropriate strategy (e.g., step-by-step logical reasoning, constructing a decision tree, or invoking an external calculator); second, the model executes the chosen strategy to produce a solution. This approach systematically embeds human-like metacognitive planning into AI workflows, resulting in enhanced robustness and adaptability.

Additionally, iterative self-refinement methods (such as ERASER (Huang et al., 2024) and Reflexion (Shinn et al., 2023)) are formalized within the scaffold as cyclic meta-instructions that prompt the model to self-assess and revise its outputs based on feedback. These techniques instantiate a reasoning loop governed by the sequence: *Solve* → *Evaluate* → *Reflect* → *Revise*, allowing the system to improve its answers over multiple iterations.

Multi-agent orchestration is another key structural element of MetaScaffolds. We leverage formal agent coordination methods—notably behavior trees (Kelley, 2024)—to coordinate specialized LLM agents and external tools in complex reasoning workflows. Behavior trees provide explicit sequencing, decision points, and fallback mechanisms to ensure robust collaboration among agents. Within a MetaScaffold, multi-agent orchestration can be flexibly combined with meta-prompting: the decision engine can assign sub-tasks to different agents or tools and integrate their results, managing roles and interactions dynamically according to a predefined coordination schema.

An applied example of the MetaScaffold framework in action, specifically tailored for a text generation task, is illustrated in Fig. 2. In this illustration, a specialized decision engine preprocesses a user query, retrieves relevant meta-prompts, and integrates contextual data along with external parameters to dynamically construct a comprehensive meta-instruction for the LLM.

Approaches like MetaRAG (Zhou et al., 2024) further demonstrate the MetaScaffold's capability to manage external knowledge integration in a metacognitive manner—adding a layer of self-monitoring control over retrieval processes. By ensuring that information gathering is context-aware and strategically planned, the scaffold remains sensitive to the environment and task-specific constraints at each step.

Collectively, the formalized MetaScaffold paradigm provides a robust, standardized schema that integrates meta-level reasoning strategies, iterative refinement loops, multi-agent orchestration, and context-aware retrieval mechanisms into a unified, interoperable framework. This paradigm is suitable for a wide range of AI reasoning and decision-making tasks, aiming to improve modularity, reusability, and reliability across domains.

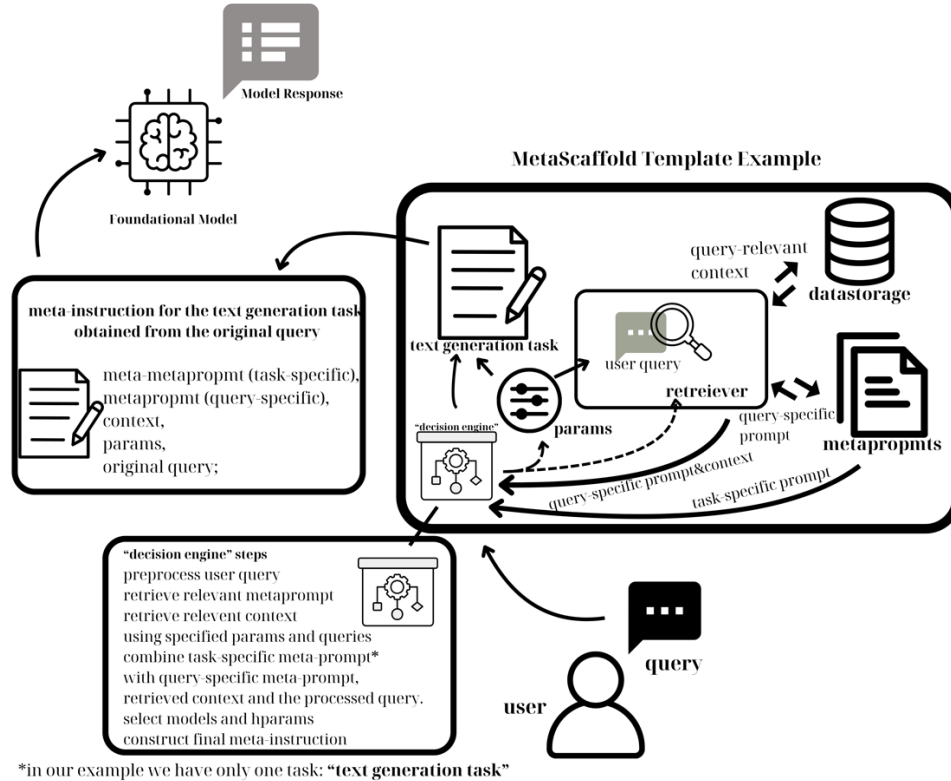


Figure 2 Example instantiation of a MetaScaffold for a text generation task.

The MSTFT Framework: Design Principles, Implementation, and Capabilities

MSTFT is a practical instantiation of the MetaScaffold paradigm, developed to provide a standardized, flexible, and modular framework for taxonomic processing tasks leveraging LLMs.

Design Principles

The MSTFT framework embodies several core design principles:

- **Modularity and Flexibility:** MSTFT's architecture is modular, enabling easy reconfiguration and integration of different reasoning components and external tools. Users can adapt or extend the framework's functionality to meet specific domain requirements without significant restructuring of the overall system.
- **Meta-Level Reasoning:** Central to MSTFT is the integration of meta-prompting and recursive meta-prompting techniques, which provide dynamic refinement and self-improvement loops for complex taxonomic reasoning tasks. This ensures that the framework can iteratively improve its output by reflecting on and adjusting its reasoning strategy.
- **Agent-Driven Orchestration:** MSTFT leverages a structured multi-agent orchestration approach, clearly defining roles and interactions between components through the meta-level decision engine. Dedicated sub-agents or modules handle distinct tasks (such as concept extraction or validation), and the decision engine coordinates them, ensuring robust handling of complex workflows involving multiple reasoning agents.
- **Semantic Consistency and Validation:** MSTFT incorporates semantic validation mechanisms by utilizing graph-based representations of taxonomies (e.g., exporting to RDF/OWL formats). This helps ensure logical consistency and integrity of the constructed taxonomic structures, as the framework can detect and correct contradictions or lapses in the hierarchy.
- **Interoperability and Standardization:** The framework is designed to be model-agnostic, allowing seamless integration with various foundational LLMs, external configuration files, and tool APIs. This promotes standardized, reproducible practices in AI-driven taxonomy processing and makes MSTFT adaptable to new models or tools with minimal changes.

Implementation Strategies

The MSTFT architecture centers around the meta-level decision engine, which is responsible for selecting and orchestrating various components—including prompts, contextual data, model parameters, tools, and memory. MSTFT follows a structured procedure to manage user queries, dynamically generating task-specific meta-instructions through the following steps:

1. **Preprocessing and Retrieval:** The user's query is first analyzed and preprocessed to extract its key topics or intents. Based on this analysis, MSTFT retrieves relevant contextual information and appropriate meta-prompts (from a library of prompt templates or previous interactions) that will guide the subsequent reasoning.
2. **Meta-Prompt Assembly:** Next, the framework assembles a task-specific meta-prompt. It combines the retrieved meta-prompt templates with the details of the current query and context to create a refined, comprehensive instruction tailored to the taxonomic reasoning task at hand. This may include injecting domain-specific guidelines or constraints into the prompt.
3. **Dynamic Model Selection:** Based on the requirements of the task, MSTFT can dynamically select the optimal language model (or configuration) to use for generation. For example, it might choose a larger, more powerful model for a particularly complex query, or a specialized model fine-tuned on taxonomic data for domain-specific terminology. It also adjusts relevant hyperparameters (such as temperature or decoding strategy) to best suit the current task.
4. **Iterative Reasoning and Validation:** The selected model produces an output (e.g., a proposed taxonomy update or classification). MSTFT then evaluates this output using its validation mechanisms—checking for semantic consistency, completeness, or errors. Identified issues trigger recursive meta-prompting: the framework formulates a follow-up prompt (or set of prompts) that asks the model to refine or correct its previous output. This loop of generation and feedback-driven refinement repeats until the taxonomy results meet predefined quality criteria or no further improvements are identified.

Results

Capabilities and Demonstrative Implementation

MSTFT demonstrates significant improvements in the clarity, precision, and adaptability of automatically constructed taxonomies. The iterative nature of its reasoning process has proven effective in reducing inaccuracies commonly associated with LLM-generated taxonomies, such as hallucinated category relationships or irrelevant branches. By repeatedly self-critiquing and refining its outputs, MSTFT progressively expands and purifies taxonomic hierarchies, ensuring that the final structures are both comprehensive and semantically coherent.

Additionally, MSTFT supports exporting the taxonomies it constructs into standard ontology formats (such as OWL). This capability enhances compatibility with external knowledge systems and tools—for instance, the resulting taxonomy can be imported into ontology editors or reasoning engines for further use—demonstrating the framework's practical interoperability.

Empirical Experiments and Validation

Experimental Setup

Objective: The primary objective of these experiments is to empirically evaluate the effectiveness of MSTFT in constructing **deep, coherent, and semantically structured taxonomies**, comparing its performance against two baseline methods: Single-Pass LLM (SP-LLM) and Iterative Prompting (IP) without meta-level orchestration.

Domains and Datasets: To evaluate generalizability, experiments were conducted in several representative domains, such as *Software, Wood, Biology, and Disease*.

For each domain, a **reference taxonomy** (gold standard) was generated using the following systematic procedure:

1. Automated extraction of hypernym–hyponym relations from WordNet (Miller, 1992).
2. Automated retrieval of Derived Terms from ConceptNet (Speer, Chin, & Havasi, 2017).
3. *Manual filtering* to eliminate overly generic or irrelevant terms, refining each taxonomy to a structured set of at least *15–20 validated concepts across three hierarchical levels*.

Taxonomies were generated via Python-based automation scripts and manually refined for accuracy to ensure robust reference standards.

Methods Compared: Three automated taxonomy construction methods were evaluated:

1. **Single-Pass LLM (SP-LLM):** Constructs a complete taxonomy in a single, non-iterative LLM invocation.
2. **Iterative Prompting (IP):** Builds taxonomies iteratively via multiple LLM calls, lacking structured meta-prompting or recursive refinement.
3. **MSTFT (Proposed Method):** Employs meta-level orchestration, recursive self-refinement, and multi-agent frameworks for structured taxonomy creation.

Model Configuration:

- All methods utilized GPT-4-Turbo (April 2025 version) via OpenAI API (OpenAI, 2023).
- Model parameters were fixed: Temperature = 0.2, Top-p = 0.95, Presence penalty = 0.8, Frequency penalty = 0.3.
- Each method-domain combination underwent three independent runs to ensure statistical reliability.

Controlled Evaluation and Reproducibility.

Although the scale of experiments is intentionally limited at this stage, all compared methods were evaluated under strictly controlled conditions: identical model version, fixed decoding parameters, unified prompt constraints, and repeated runs. This

design isolates the impact of orchestration strategy (single-pass vs. iterative vs. meta-level orchestration) as the primary experimental variable, reducing confounding factors related to model stochasticity or prompt formulation.

Evaluation Metrics

Initially, we considered standard automatic metrics, but quickly identified their limitations. In particular, the reference taxonomy for the "Software" domain contained only 33 concepts, including low-relevance or culturally specific terms such as 'microserf', 'microhoo', 'microshaft', and 'microsoftify'. This made pure overlap-based comparisons insufficient.

As a result, we emphasized qualitative evaluation by human reviewers, guided by the following adapted metrics:

- Conceptual Relevance: Evaluates whether generated subconcepts are meaningful, distinctive, and appropriately scoped.
- Hierarchical Coherence: Assesses logical consistency and structural integrity across levels.
- Instruction Adherence: Checks whether generated taxonomies conform to prompts, particularly in maintaining depth (three levels).
- Generation Efficiency: Compares average API cost per taxonomy.

Additionally, reviewers recorded:

- Average number of valid subconcepts per level.
- Error types, including redundancy, inconsistency, and conceptual drift.

Stability Across Runs.

Each method–domain pair was executed in three independent runs. Across runs, MSTFT demonstrated low variance in both structural depth and number of valid subconcepts ($\pm 5\%$), while SP-LLM and IP exhibited higher variance, particularly in hierarchy depth and category consistency. Although the limited number of runs does not allow formal statistical hypothesis testing, this stability trend provides an empirical proxy for robustness.

Results and Analysis

Quantitative Comparison: In the "Software" domain, MSTFT generated 80–90 well-structured and semantically distinct subconcepts, outperforming SP-LLM (~30 concepts) and IP (~50 concepts), which showed issues with redundancy and ambiguity. For example, SP included overlapping terms like *Artificial Intelligence and Machine Learning Software* and *AI Platforms and Tools*, while IP introduced vague categories such as *Word Processing Software*.

Structural Fidelity: MSTFT consistently adhered to the three-level depth constraint, producing coherent multi-level structures. In contrast, SP and IP outputs often lacked depth or grouped concepts inconsistently.

API Cost Efficiency: SP was the most economical method (~\$0.01), while IP incurred the highest cost (~\$0.10). MSTFT maintained a reasonable balance between cost (~\$0.03) and quality.

Cross-Domain Observations: MSTFT's performance generalized across other domains such as "Wood" and "Biology", producing more differentiated and structured hierarchies than SP or IP.

For detailed comparative results and example outputs in the "Software" domain, see Appendix A.

Conclusion: These results demonstrate MSTFT's ability to deliver deep, interpretable, and hierarchically aligned taxonomies with higher conceptual relevance and practical structure. Its performance justifies modestly higher generation cost and paves the way for expert-driven refinement and broader human evaluation in future work.

Importantly, the current evaluation represents only a preliminary assessment. The primary aim at this stage was to demonstrate the feasibility and potential of MSTFT as a MetaScaffolding-driven framework for taxonomy construction, rather than to conduct a definitive benchmark. While early results are promising, further large-scale experiments with diverse concept domains, deeper hierarchies, and independent expert reviews are required to validate generalizability and robustness. We also plan to integrate more systematic annotation workflows and comparative studies against existing ontology engineering tools in future research. Nonetheless, MSTFT has already shown strong potential as a generalizable and semantically grounded methodology for structured concept organization.

Error Pattern Analysis.

Qualitative error analysis revealed systematic differences between methods. SP-LLM errors were dominated by synonym redundancy and flat enumerations without functional separation. IP frequently introduced category leakage, mixing orthogonal criteria (e.g., licensing, deployment, and functionality within a single level). In contrast, MSTFT errors were rare and primarily related to borderline concept granularity rather than logical inconsistency, indicating that meta-level orchestration mainly reduces structural, not lexical, error modes.

Importantly, the goal of the present evaluation is not to establish a definitive benchmark, but to empirically demonstrate that meta-level orchestration yields systematically different and more stable structural outcomes than baseline prompting strategies under controlled conditions.

Conclusion

Conclusion and Future Research Directions

In this paper, we introduced the MSTFT, highlighting its design principles, implementation strategies, and demonstrated capabilities as a standardized scaffolding framework for taxonomy construction. MSTFT addresses critical gaps in existing methodologies by providing a flexible meta-level framework that leverages advanced reasoning techniques to improve structure and accuracy.

Future research will proceed along two directions: improving MSTFT itself and advancing the broader MetaScaffold framework.

For MSTFT, key enhancements include strengthening semantic validation via constraint-checking, graph consistency checks, and lightweight reasoning to address structural or semantic errors. We also plan broader empirical evaluations across domains and languages to test robustness and generalizability. Interfaces for expert-in-the-loop adjustments will be refined to let human reviewers guide meta-level decisions during taxonomy construction.

At the MetaScaffold level, future work will explore applications in AI reasoning tasks such as ontology alignment, long-term planning, and multi-step question answering. This includes support for multi-modal reasoning (e.g., integrating textual, visual, and tabular inputs), richer meta-prompt designs, and coordinated orchestration across agent systems. MSTFT serves here as a concrete use case, offering transferable insights for more general-purpose scaffolding systems targeting complex, knowledge-intensive workflows.

While the present study focuses on text-based taxonomies, preliminary pilot experiments indicate that the MetaScaffold approach is compatible with multimodal inputs, such as image-derived labels or tabular metadata, when these are incorporated as structured context constraints rather than raw generative targets. A systematic multimodal evaluation is planned as a separate study.

Limitations

While the MetaScaffold TaxoFrame Template (MSTFT) offers a flexible and modular approach to taxonomy construction, several limitations remain. First, the framework's reliance on large language models (LLMs) means that its performance is inherently tied to the quality and biases present in the underlying models. Second, the recursive meta-prompting mechanism, while powerful, can lead to increased computational overhead, potentially limiting scalability in resource-constrained environments. Third, the domain-agnostic design, although versatile, may not capture domain-specific nuances without additional fine-tuning or customization. Lastly, the evaluation of MSTFT has been primarily conducted in controlled settings; its effectiveness in diverse real-world applications remains to be thoroughly assessed.

Ethical Considerations

The deployment of MSTFT raises several ethical considerations. Given its dependence on LLMs, there is a risk of propagating existing biases present in the training data, which could lead to skewed or unfair taxonomic structures. Additionally, the automated nature of taxonomy generation may inadvertently overlook culturally sensitive classifications or reinforce stereotypes. To mitigate these risks, it is crucial to incorporate bias detection and correction mechanisms within the framework. Furthermore, transparency in the decision-making processes of the agents involved should be ensured to foster trust and accountability. Users should be made aware of the system's limitations and the potential need for human oversight, especially in high-stakes or sensitive domains.

Evaluation Results for the Concept Software

This appendix provides detailed comparative evaluation of taxonomy generation outputs for the concept *Software*, generated using three distinct methods: Single-Pass LLM (SP), Iterative Prompting (IP), and the proposed MSTFT framework.

1. Reference Gold Standard Overview

The reference taxonomy for "Software" was constructed from WordNet hypernyms and ConceptNet derived terms. It consists of 33 concepts. However, more than 10 entries were deemed low-value or culturally specific, e.g., '*microserf*', '*microhoo*', '*microshaft*', '*microsoftify*', '*softwarily*', '*microsoftian*'. These were excluded from qualitative assessments.

2. Conceptual Relevance (Qualitative Judgement)

- SP: 30 subconcepts; generally meaningful, though sometimes overlapping (e.g., '*AI Platforms and Tools*' vs '*Artificial Intelligence and Machine Learning Software*').
- IP: 50+ subconcepts; included useful terms but often introduced blurry categories (e.g., '*Word Processing Software*' as an isolated domain).
- MSTFT: 80–90 subconcepts; high conceptual quality, well-differentiated functional, structural, and performance-related branches.

3. Hierarchical Coherence

- SP: Flat or ambiguous hierarchy, lacking clear structuring across multiple levels.
- IP: Shallow depth with inconsistent grouping (e.g., deployment models mixed with licensing).
- MSTFT: Strong multi-level structure (3+ levels), including nested categories such as: *Functionality Classification* → *Operating Systems* → *Desktop Operating Systems*

4. Instruction Adherence (Three-level depth)

- SP: Partial adherence; mostly two-level or flat taxonomies.
- IP: Similar to SP; depth occasionally broken or inconsistent.
- MSTFT: Consistent adherence to prompt constraint of 3 levels.

5. Average Number of Valid Subconcepts per Level (Estimated)

- SP: ~20.
- IP: ~15–20 per level, though with redundancy.
- MSTFT: ~25–30 per level across three levels, minimal duplication.

6. Error Types Observed

- SP: Redundant terms (e.g., synonyms), occasional under-specification.
- IP: Redundancy and inconsistency in groupings.
- MSTFT: Near-zero logical errors; structure aligned with instructions.

7. API Cost per Full Taxonomy (Estimated)

Approximate API costs per full taxonomy are: SP (~\$0.01), IP (~\$0.10), and MSTFT (~\$0.03), indicating that MSTFT remains both effective and cost-efficient.

8. Summary Judgement

MSTFT demonstrated superior performance across all axes: depth, clarity, consistency, and taxonomic richness. It achieved this at moderate cost, while baseline methods struggled with either depth (SP) or structural coherence (IP).

Author's contribution. Danylo Dvoichenkov – conceptualization and development of the MetaScaffold paradigm and MSTFT framework, methodology design, literature review, experimental implementation, data collection and analysis, manuscript writing and revision. Oleksandr Marchenko – research supervision, critical evaluation and quality assessment, methodological guidance, identification of relevant literature, and manuscript review with constructive feedback.

Funding sources. This study did not receive any grant from a financial institution in the public, commercial, or non-profit sectors.

Disclosure of Delegation to Generative AI. The authors declare the use of generative AI in the research and writing process. According to the GAIDeT taxonomy (2025), the following tasks were delegated to GAI tools under full human supervision:

- Evaluation of the novelty of the research and identification of gaps;
- Reformatting;
- Quality assessment;
- Identification of limitations;
- Recommendations.

The GAI tool used was: chatgpt 4.5, claude sonnet 4.5. Responsibility for the final manuscript lies entirely with the authors. GAI tools are not listed as authors and do not bear responsibility for the final outcomes.

References

- Bhutani, N., Kandogan, E., Rahman, S., Mitchell, T., Hruschka, E., & Pezeshkpour, P. (2024). *Reasoning capacity in multi-agent systems: Limitations, challenges and human-centered solutions* (arXiv:2402.01108). arXiv. <https://doi.org/10.48550/arXiv.2402.01108>
- Gao, P., Xie, A., Mao, S., Wu, W., Xia, Y., Mi, H., & Wei, F. (2024). *Meta reasoning for large language models* (arXiv:2406.11698). arXiv. <https://doi.org/10.48550/arXiv.2406.11698>
- Huang, X., Cheng, S., Huang, S., Shen, J., Xu, Y., Zhang, C., & Qu, Y. (2024). *QueryAgent: A reliable and efficient reasoning framework with environmental feedback-based self-correction* (arXiv:2403.11886). arXiv. <https://doi.org/10.48550/arXiv.2403.11886>
- Kelley, R. (2024). *Behavior trees enable structured programming of language model agents* (arXiv:2404.07439). arXiv. <https://doi.org/10.48550/arXiv.2404.07439>
- Liu, Y., Lo, S. K., Lu, Q., Zhu, L., Zhao, D., Xu, X., Harrer, S., & Whittle, J. (2024). *Agent design pattern catalogue: A collection of architectural patterns for foundation model based agents* (arXiv:2405.10467). arXiv. <https://doi.org/10.48550/arXiv.2405.10467>
- Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., Wen, J.-R., & Wang, L. (2024). *A survey on large language model based autonomous agents*. *Frontiers of Computer Science*. 18, 186345. <https://doi.org/10.1007/s11704-024-40231-1>
- Miller, G. A. (1992). *WordNet: A lexical database for English*. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23–26, 1992*. ACL Anthology. <https://aclanthology.org/H92-1116/>
- OpenAI. (2023). *GPT-4 API general availability*. <https://openai.com>
- Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., & Yao, S. (2023). *Reflexion: Language agents with verbal reinforcement learning*. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS 2023)*. Curran Associates, Inc. <https://doi.org/10.48550/arXiv.2303.11366>
- Speer, R., Chin, J., & Havasi, C. (2017). *ConceptNet 5.5: An open multilingual graph of general knowledge*. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI '17)* (pp. 4444–4451). Association for the Advancement of Artificial Intelligence. <https://doi.org/10.1609/aaai.v31i1.11164>
- Suzgun, M., & Kalai, A. T. (2024). *Meta-prompting: Enhancing language models with task-agnostic scaffolding* (arXiv:2401.12954). arXiv. <https://doi.org/10.48550/arXiv.2401.12954>
- Yao, S., Zhao, J., Yu, D., Du, N., Shafraan, I., Narasimhan, K., & Cao, Y. (2023). *ReAct: Synergizing reasoning and acting* (arXiv:2210.03629). arXiv. <https://doi.org/10.48550/arXiv.2210.03629>
- Zhang, Y., Yuan, Y., & Yao, A. C.-C. (2024). *Meta prompting for AI systems* (arXiv:2311.11482). arXiv. <https://doi.org/10.48550/arXiv.2311.11482>
- Zhou, Y., Liu, Z., Jin, J., Nie, J.-Y., & Dou, Z. (2024). *Metacognitive retrieval-augmented large language models* (arXiv:2402.11626). arXiv. <https://doi.org/10.48550/arXiv.2402.11626>

Отримано редакцією журналу / Received: 30/08/2025

Прорецензовано / Revised: 31/10/2025

Схвалено до друку / Accepted: 29/12/2025

Данило ДВОЙЧЕНКОВ, аспірант

ORCID ID: 0009-0007-1935-6743

e-mail: kmwp@irtc.org.ua

Інститут інформаційних технологій та систем Національної Академії Наук України, Київ, Україна

Олександр МАРЧЕНКО, д-р фіз.-мат. наук, проф.

ORCID ID: 0000-0002-5408-5279

e-mail: omarchenko@univ.kiev.ua

Інститут інформаційних технологій та систем Національної Академії Наук України, Київ, Україна

Київський національний університет імені Тараса Шевченка, Київ, Україна

ДО ВИЯВЛЕННЯ ГЛОБАЛЬНОГО СТАНДАРТУ МЕТАСКАФOLDИНГУ: ВИПАДОК ШАБЛОНУ METASCAFFOLD TAXOFRAME ЯК ПАРАДИГМАТИЧНА ДЕМОНСТРАЦІЯ ПІДХОДУ

Вступ. Поширення AI-систем на основі великих мовних моделей (LLM) призвело до широкого впровадження каркасних методологій міркування, які покращують можливості моделей через структуровані підходи, включаючи ланцюгові промпти, мульти-агентну координацію та інтеграцію зовнішніх інструментів. Однак спонтанний характер сучасних процесів проектування каркасів створює значні виклики щодо сумісності, масштабованості та відтворюваності в різних AI-робочих процесах. Ця стаття представляє парадигму МетаСкафолдингу – абстракцію вищого рівня для структурування та оркестрації процесів міркування AI, яка забезпечує багаторазові, систематичні, домен-агностичні принципи для організації знань та кроків міркування. Ми представляємо Шаблон MetaScaffold TaxoFrame (MSTFT) як практичну реалізацію цієї парадигми, спеціально розроблену для завдань таксономічної обробки, демонструючи, як стандартизовані мета-структури можуть покращити модульність та ефективність AI-систем, зберігаючи гнучкість у різних доменах та застосуваннях.

Методи. Ми формалізуємо metascaffold як структуровані механізми оркестрації, які динамічно інтегрують мета-рівневий промптинг, агент-орієнтоване міркування, цикли саморефлексії та зовнішнє отримання знань через центральний механізм прийняття рішень. Фреймворк MSTFT реалізує цю парадигму з основними компонентами, включаючи: (1) мета-промпти та рекурсивні мета-промпти для ітераційних циклів удосконалення, (2) модульні агентні системи для спеціалізованих можливостей міркування, (3) управління пам'яттю та контекстом для семантичної послідовності, та (4) налаштовувані параметри моделі для оптимізації завдань. Наша реалізація відповідає структурованій процедурі, що включає попередню обробку та отримання, збирання мета-промптів, динамічний вибір моделі та ітераційне міркування з валідацією. Ми оцінили MSTFT порівняно з базовими методами Single-Pass LLM (SP-LLM) та Iterative Prompting (IP) у кількох доменах (програмне забезпечення, деревина, біологія, хвороби), використовуючи референсні таксономії, побудовані з гіпероніміє WordNet та похідних термінів ConceptNet, з фокусом оцінки на концептуальній релевантності, ієрархічній когерентності, дотриманні інструкцій та ефективності генерації.

Результати. Експериментальна валідація демонструє перевершуючу ефективність MSTFT за всіма метриками оцінки. У домені програмного забезпечення MSTFT зазвичай генерує 80-90 добре структурованих та семантично відмінних підконцептів порівняно з SP-LLM (~30 концептів) та IP (~50 концептів), які демонстрували проблеми з надмірністю та неоднозначністю. MSTFT послідовно підтримував трирівневу ієрархічну глибину згідно зі специфікацією, тоді як базові методи продукували плоскі або непослідовні структури. Фреймворк досяг оптимального балансу між якістю та ефективністю витрат (~\$0.03 за таксономію) порівняно з SP-LLM (~\$0.01) та IP (~\$0.10). Кроссдоменна оцінка підтвердила узагальнюваність MSTFT, продукуючи більш диференційовані та структуровані ієрархії в доменах деревини, біології та хвороб. Якісний аналіз виявив, що виводи MSTFT демонструють вищу концептуальну релевантність, ієрархічну когерентність та мінімальні логічні помилки з приблизно 25-30 валідними підконцептами на рівень на три ієрархічні рівні.

Висновки. Ця робота представляє парадигму МетаСкафолдингу та демонструє її практичну реалізацію через MSTFT, вирішуючи критичні прогалини в існуючих методологіях AI-каркасів. MSTFT успішно інтегрує мета-рівневе міркування, ітераційне удосконалення та мульти-агентну оркестрацію в уніфікований, сумісний фреймворк, що значно покращує якість конструювання таксономій зберігаючи обчислювальну ефективність. Модульний дизайн фреймворку, механізми семантичної валідації та стандартизований підхід забезпечують основу для ширших застосувань у завданнях AI-міркування, включаючи вирівнювання онтологій, довгострокове планування та багатокрокове відповідання на запитання. Майбутні напрями досліджень включають покращену семантичну валідацію, підтримку мульти-модального міркування та розширену емпіричну оцінку в різних доменах та мовах. Хоча існують обмеження щодо залежностей від LLM та обчислювальних накладних витрат, MSTFT встановлює перспективну парадигму для стандартизованих, багатофункціональних AI-каркасних фреймворків, які балансують гнучкість із систематичною структурністю.

Ключові слова: MetaScaffold, конструювання таксономій, мета-промптинг, мульти-агентні системи, великі мовні моделі, фреймворки AI-міркування, організація знань, ітераційне удосконалення, онтологічна інженерія.

All authors have read and approved the final version of the manuscript. The authors declare no conflicts of interest. One of the co-authors is a member of the editorial board of the publication, therefore he did not participate in the review and decision-making regarding the publication of this article. The funders had no involvement in the study design; the collection, analysis, or interpretation of data; the preparation of the manuscript; or the decision to publish the results.

Усі автори ознайомилися з рукописом і схвалили його остаточну версію. Автори заявляють про відсутність конфлікту інтересів. Один із співавторів є членом редколегії видання, тому не брав участі у рецензуванні та прийнятті рішення щодо публікації цієї статті. Джерела фінансування не були залучені до розроблення дизайну дослідження; збору, аналізу чи інтерпретації даних; підготовки рукопису; або ухвалення рішення про публікацію результатів.