2026 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
DIGITAL ENGINEERING/SYSTEMS ENGINEERING (DE/SE) TECHNICAL SESSION
AUGUST 11-13, 2026 - NOVI, MICHIGAN

An Agentic AI Vision for Combat Vehicle Architecture Model

Macam Dattathreya, PhD

US Army DEVCOM GVSC

ABSTRACT

This position paper presents *a vision for reforming combat vehicle architecture model development through Agentic AI.* *Manual Model Based Systems Engineering (MBSE) approaches create critical bottlenecks in embedding modularity, cybersecurity, safety, and interoperability from inception, resulting in inconsistent architecture and extended development cycles. The proposed conceptual framework leverages specialized AI agents and large language models to automate modular system model generation. This vision addresses an unexplored integration opportunity by combining mature MBSE tools, Modular Open Systems Approach (MOSA) frameworks, and multi-agent AI into cohesive, standards-based automation. The paper presents conceptual multi-agent architecture with dynamic validation techniques, identifies technical challenges and coordination complexity, and proposes a phased research agenda with success criteria. Realizing this vision could reduce design timelines from month to week while ensuring consistent MOSA compliance across the combat vehicle fleet.*

Citation: M. Dattathreya, "An Agentic AI Vision for Combat Vehicle Architecture Model," In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 11-13, 2026.

1. INTRODUCTION

Modernizing combat vehicle systems using a Modular Open Systems Approach (MOSA) [1] and Model-Based Systems Engineering (MBSE) [2] is hampered by the difficulty of embedding modularity [3], cybersecurity, and interoperability from the initial design stages. Traditional system architecture modeling is manual [4][5], slow, and prone to errors, often requiring months of work and significant rework. This paper proposes a new, automated framework to address these issues.

1.1. The Modernization Challenge

Current methods for developing combat vehicle architecture models are inefficient. Systems engineers manually translate MOSA principles into complex Systems Modeling Language (SysML) models [4][5]. This process is not only time-consuming, taking 4-6 months per platform, but also results in a 30-40% rework rate when security and safety issues are discovered late in development.

These traditional approaches are inadequate for the complexity of modern combat systems, which integrate numerous electronic, software, and network components with strict performance and survivability requirements.

1.2. A New Agentic AI Framework

While established standards exist, current practices often rely on manual interpretation and implementation, leading to inconsistent architecture and extended development cycles. Moreover, no existing frameworks effectively integrate AI-driven automation with standards-based MBSE automation to address these challenges comprehensively

This paper introduces a forward-thinking framework that uses advanced artificial intelligence to automate the creation of SysML architecture models while applying MOSA principles throughout the design process. By employing specialized AI agents [6], this framework can optimize modularity, cybersecurity, safety, and interoperability throughout the design process. This "shift-left" approach, which integrates critical security and safety considerations at the earliest stages, aligns with the Department of War's modernization goals.

Unlike generic AI tools, this framework creates regulation-aware, domain-specific systems that automate formal system architecture modeling, resolving a major bottleneck in systems engineering.

1.3. Contribution and Scope

This work offers a vision for the future of defense systems engineering, arguing for the necessity of Agentic AI to overcome the limitations of manual MBSE. The key contributions are:

1. A conceptual multi-agent AI architecture for MOSA-guided system design.
2. Identification of technical challenges in automating SysML model generation.
3. Innovative validation methods to reduce reliance on manual review.

Recent advances in Large Language Models (LLM)s, now enable formal reasoning and multi-step problem solving. Combined with mature MBSE tools offering API access, standardized MOSA frameworks like Ground Combat Systems Common Infrastructure Architecture (GCIA) [7], and proven multi-agent coordination techniques, these independently developed technologies can now be integrated representing this work's novel contribution.

This paper is a position statement intended to spark discussion and guide future research. It does not present a completed or validated system but instead proposes a research agenda to explore the feasibility of this transformative approach.

The remainder of the paper is organized as follows: Section 2 reviews related work, Section 3 presents the proposed conceptual framework, Section 4 discusses technical challenges, and Section 5 outlines the research agenda. Section 6 concludes the paper.

2. THE LITERATURE REVIEW RESEARCH GAP

A review of current technologies reveals a significant integration gap in defense systems engineering. While MBSE tools, multi-agent AI systems [6], and LLMs [8] have all matured independently, no existing work combines them into a cohesive framework for automated, standards-based system architecture model generation.

Current MBSE tools provide sophisticated modeling environments but are fundamentally manual, requiring human input for design decisions and lacking the ability to automatically enforce MOSA principles embedded in the frameworks like GCIA. Likewise, multi-agent AI systems have proven successful in other industries [8] but have not been adapted for the specific cybersecurity, modularity, and validation requirements of defense applications. Finally, while recent LLMs show promise, they remain generic and lack the domain-specific knowledge required to generate the formal, compliant SysML models needed for military systems.

This research position paper addresses this gap by proposing a vision for specialized agentic AI framework that integrates domain knowledge with collaborative AI reasoning to automate the generation of GCIA-guided SysML models. The novelty of this approach lies in the systematic integration of these technologies, creating domain-aware AI agents enhanced with advanced reasoning to solve a critical bottleneck in defense systems engineering.

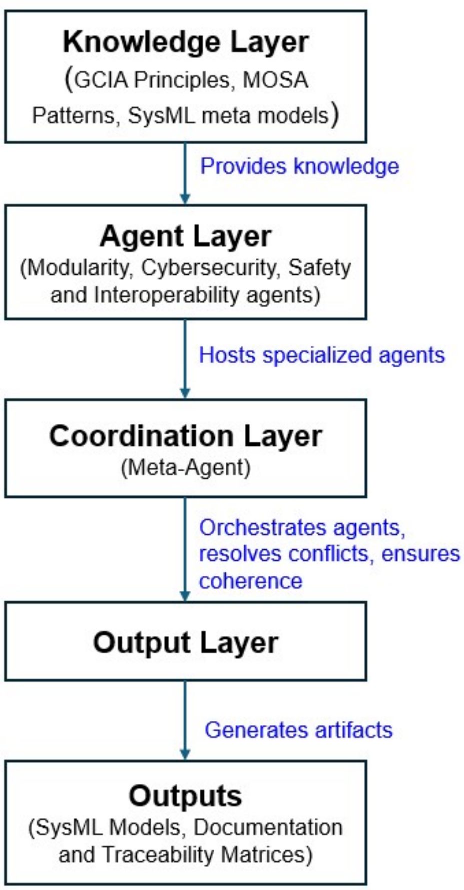

Figure 1: Four-Layer Architecture Overview: Shows the hierarchical structure of the framework, from knowledge representation to final outputs.

3. PROPOSED CONCEPTUAL FRAMEWORK

The conceptual architecture of the proposed framework as depicted in **Figure 1**, and the subsections are a description intentionally at high-level, focusing on the intended structure to illustrate the vision and enable discussion of feasibility and challenges. Detailed algorithms, implementation specifics, and performance characteristics are subjects for future research and will be developed during the proof-of-concept phase.

3.1. Architecture Overview

The framework transforms high-level requirements into MOSA/GCIA-aligned SysML models through four integrated layers:

Knowledge Layer: Machine-readable representations of GCIA principles, MOSA patterns, SysML metamodels, and combat vehicle domain knowledge. Requires significant knowledge engineering to formalize implicit expert knowledge and design rationale not captured in existing knowledge artifacts.

Agent Layer: Four specialized agents address distinct, often-conflicting architectural concerns aligned with GCIA principles:

- **Modularity Agent:** System decomposition, functional allocation, and MOSA-compliant interfaces. Some example behaviors include decomposing systems into cohesive modules and minimizing coupling.
- **Cybersecurity Agent:** Defense-in-depth, trust boundaries, and threat modeling. Examples of behavior include ensuring defense-in-depth and applying GCIA cybersecurity infrastructure principles.

- **Safety Agent:**
  Hazard analysis, safety mechanisms, and fail-safe behaviors such as validating against single point failures. Applying GCIA safety infrastructure principles.
- **Interoperability Agent:**
  Standards compliance and system compatibility such as validating standards conformance with GCIA, VICTORY, CMOSS, FACE.

This design balances architectural coverage with coordination complexity and will be validated during proof-of-concept development.

Coordination Layer: A meta-agent orchestrates specialized agents, manages interactions, and resolves conflicts through structured negotiation without making architectural decisions.

Output Layer: Generates SysML models in the XML Metadata Interchange (XMI) format, documentation, and traceability matrices linking decisions to requirements and GCIA principles.

3.2. Agent Architecture and Reasoning

Each specialized agent comprises four functional modules:

Perception Module analyzes requirements, constraints, and current model state to identify relevant architectural concerns.

Knowledge Base contains domain-specific rules, design patterns, and heuristics:

- *Modularity Agent*: Coupling metrics, cohesion principles, interface patterns
- *Cybersecurity Agent*: Threat models, security patterns such as defense-in-depth, least privilege.
- *Safety Agent*: Hazard taxonomies, safety patterns such as redundancy, fail-safe defaults.
- *Interoperability Agent*: Standards specifications, interface protocols, compatibility rules.

Reasoning Engine employs LLM-based reasoning augmented with constraint-based optimization through: (1) constraint formulation from architectural principles, (2) LLM-generated candidate designs, (3) formal rule validation, and (4) iterative refinement if constraints are violated. This hybrid approach constrains LLM outputs to valid SysML constructs and uses rule-based validation for critical decisions.

Action Module proposes specific SysML model elements such as blocks, interfaces, constraints with justification and traceability.

3.3. Coordination Workflow

The meta-agent coordinates specialized agents through a five-phase workflow. **Figure 2** illustrates this process as a flowchart.

Phase 1: Parallel Analysis - All agents independently analyze requirements and propose architectural elements. This parallel approach ensures each concern is fully represented before negotiation begins**.**

Phase 2: Conflict Identification - The meta-agent detects incompatibility between agent proposals. Common conflicts include:

- Security isolation requirements vs. interoperability needs
- Safety redundancy vs. modularity and cost constraints
- Standardized interfaces vs. optimized custom designs

Phase 3: Negotiation - Agents engage in structured dialogue to resolve conflicts. The negotiation process uses multi-objective optimization principles:

- Each agent quantifies the impact of proposed compromises on its objectives
- The meta-agent searches for Pareto-optimal solutions where no agent can improve without harming another
- Trade-off decisions are documented with explicit rationale for human review

What happens if agents cannot reach consensus? The current vision proposes

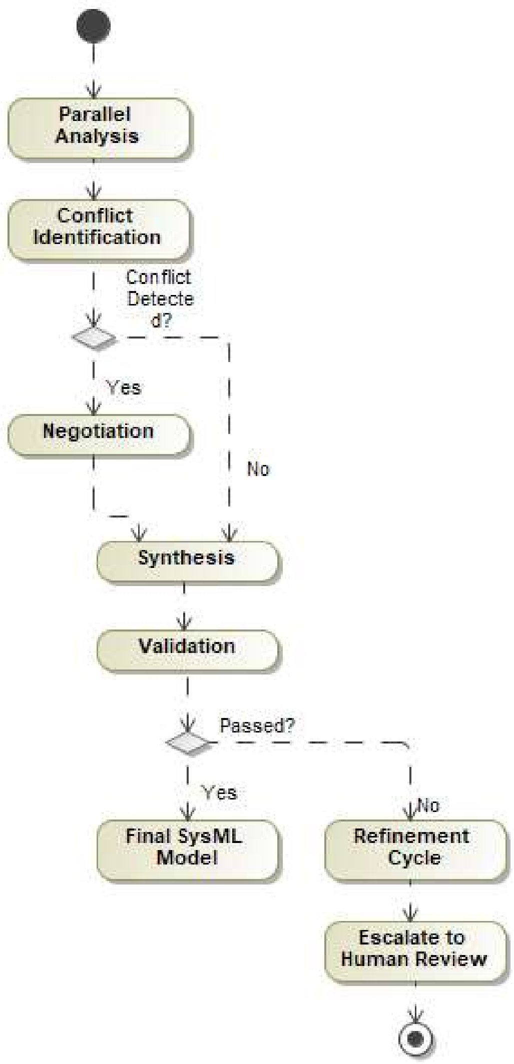

Figure 2: The coordination workflow showing the five phases: Parallel Analysis, Conflict Identification, Negotiation, Synthesis, and Validation, with iterative refinement and escalation paths for unresolved issues.

escalating unresolved conflicts to human systems engineers or subject matter experts for adjudication. However, this reintroduces manual bottlenecks. Future research must develop principled methods for automated trade-off decisions or clearly define when human judgment is essential.

Phase 4: Synthesis - The meta-agent integrates resolved proposals into a coherent SysML model, ensuring consistency across all model views (structural, behavioral, parametric); Completeness of interfaces and dependencies; Traceability from requirements through design decisions

Phase 5: Validation - The complete architecture is verified against:

- GCIA/MOSA compliance checklists
- Requirements coverage and traceability
- Internal consistency i.e., no conflicting constraints
- Completeness i.e., all required elements present)

If violations are detected, the meta-agent triggers additional reasoning cycles with specific feedback to relevant agents. To prevent infinite loops, a maximum iteration count is enforced, after which unresolved issues are flagged for human review.

3.4. Model Generation

The framework generates SysML architecture models through progressive elaboration. Requirements Modeling creates hierarchical requirement diagrams with traceability. Structural Modeling produces block definition diagrams, internal block diagrams, and package diagrams. Behavioral Modeling captures workflows through activity diagrams, communication through sequence diagrams, and component behaviors through state machines. Parametric Modeling encodes constraints and performance requirements. Dynamic validation techniques are integrated throughout the model generation process, including AI-driven simulations to test architectural robustness and real-time compliance checking against GCIA/MOSA principles. These methods reduce reliance on manual expert review and improve trust in AI-generated architectures. The innovation captured in [4] and [5] can be automated in this framework.

4. TECHNICAL CONSIDERATIONS

This section identifies key obstacles and discusses potential solution approaches without claiming to have fully resolved them. These challenges represent the core research questions that must be addressed to validate the feasibility of this approach.

4.1. Knowledge Base Development

The GCIA/MOSA framework provides a structured foundation for AI by documenting principles in the SysML modeling language. However, this structure alone is insufficient, as it lacks the critical context behind the design. Key information such as the rationale for engineering choices, the criteria for decisions, and the implicit knowledge of the experts is not captured in the models.

To address this gap, the framework must be enhanced by:

1. Developing formal ontologies or data models to supplement the existing models with a layer of reasoning rules and constraints.
2. Extracting reusable design patterns and architectural principles from the current models to apply them in future efforts.
3. Collaborating with experts in an iterative process to capture and integrate their tacit, real-world knowledge into the knowledge representation.

4.2. LLM Limitations

Current LLMs exhibit limitations including hallucination, inconsistency, and difficulty with formal reasoning that could compromise architecture quality. Mitigation strategies include constrained generation limiting outputs to valid SysML constructs through grammar-based constraints, multi-step reasoning breaking complex decisions into verifiable steps, hybrid reasoning combining LLMs with rule-based systems for critical safety and security decisions, and human-in-the-loop validation at key architectural decision points. Additionally, the framework incorporates neuro-symbolic reasoning [9] to enhance formal reasoning capabilities, enabling LLMs to handle complex architectural trade-offs more effectively. Reinforcement learning is employed to iteratively improve agent performance based on feedback from validation processes.

4.3. Agentic Complexity

Agentic AI is a very new concept. Coordinating multiple specialized agents with potentially conflicting objectives presents significant challenges. The framework must balance competing concerns such as security measures that may impede modularity or safety redundancy that complicate interfaces. Solutions include developing sophisticated negotiation protocols using multi-objective optimization principles, establishing clear priority hierarchies for resolving conflicts, and implementing validation mechanisms to ensure architectural coherence across agent contributions.

4.4. Validation and Trust

Ensuring AI-generated architectures is correct and GCIA/MOSA-aligned requires rigorous validation to establish trust in safety-critical defense applications. Approaches include automated compliance checking against GCIA principles and MOSA requirements, structured expert review by systems engineers to validate design decisions, and maintaining complete traceability documentation linking all architectural elements to requirements and GCIA/MOSA principles. Incremental adoption starting with the framework as an assistant tool before increasing autonomy allows trust to build through demonstrated reliability.

5. FUTURE IMPLICATIONS

The proposed framework has significant implications for Army combat vehicle development. Automating architecture modeling could reduce design timelines from months to weeks, enabling faster response to emerging threats. AI-driven generation ensures consistent application of MOSA/GCIA principles across programs, improving interoperability across the combat vehicle fleet. Automated documentation provides unprecedented traceability from requirements through implementation, supporting verification and future modernization. The ability to rapidly generate alternative architecture models enables more comprehensive design space exploration, supporting better-informed acquisition decisions.

Realizing this vision requires a phased research program. Near-term research should focus on proof-of-concept implementation demonstrating single-agent capabilities for one architectural concern, validation methodology establishing metrics for evaluating AI-generated architectures, and user studies understanding how systems engineers interact with AI-assisted tools. Mid-term research should address multi-agent coordination mechanisms, conflict resolution algorithms for competing architectural concerns, and tool integration with major MBSE platforms. Long-term research should pursue autonomous architecture optimization for multiple objectives, learning from experience to improve agent performance, and generalization to other defense domains.

Several challenges could impede realization of this vision. Technical risks include LLM capabilities proving insufficient for formal reasoning required in safety-critical architecture and coordination complexity growing exponentially with system size. Organizational risks involve cultural resistance to AI-generated designs in conservative defense acquisition environments and difficulty obtaining access to classified system information for realistic validation. Policy risks include evolving regulations around AI use in defense systems and intellectual property questions regarding AI-generated designs. Addressing these risks requires sustained collaboration among researchers, practitioners, acquisition professionals, and policymakers.

Additionally, dynamic validation techniques, such as AI-driven simulations, enable more robust testing of architecture, reducing risks associated with safety-critical systems.

The proposed criteria for evaluating progress toward this vision include the following: Proof-of-concept demonstrating single-agent SysML generation; validation framework accepted by systems engineering community; multi-agent coordination producing coherent architectures; expert evaluation showing AI-generated models meet quality standards; Adoption by at least one pilot program.

6. CONCLUSIONS

This paper presents a vision to revolutionize combat vehicle design by automating architecture modeling with Agentic AI. The proposed framework uses specialized AI agents, guided by MOSA principles, to overcome the delays and inconsistencies of traditional systems engineering.

By orchestrating these agents and using AI-driven simulations for validation, this approach drastically reduces design timelines from months to weeks while ensuring robust, compliant architectures. While challenges exist, this work establishes a research foundation to enable faster military modernization and sets the stage for the future of AI-assisted systems engineering in both defense and commercial domains.

7. REFERENCES

[1] Office of the Under Secretary of Defense for Acquisition and Sustainment, "Modular Open Systems Approach (MOSA) Implementation Guidebook,"

Version 1.1, U.S. Department of Defense, Washington, D.C., 2019.

[2] F, S., Moore, A., & Steiner, R. (2014). A Practical Guide to SysML: The Systems Modeling Language. Morgan Kaufmann.

[3] J. A. Z-M. Gao, "A Systems Engineering Perspective on the Business Case for Modular Open Systems Architecture," *IEEE Systems Journal*, vol. 14, no. 3, pp. 4075-4083, Sept. 2020.

[4] M. Dattathreya, “A Comprehensive Modularity Framework for Military Ground Vehicle Architecture Models,” In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 12-14, 2025.

[5] M. Dattathreya, “The art of open-standard system architecture for military ground vehicles,” In Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), NDIA, Novi, MI, Aug. 13-15, 2024.

[6] M. Wooldridge, “An *Introduction to MultiAgent Systems,”* Wiley, 2009.

[7] NAMC consortium URL, https://www.namconsortium.org/article/news/ground-combat-systems-gcs-common-infrastructure-architecture-gcia-community.

[8] L. Weng et al., "A Survey on Large Language Model based Autonomous Agents," arXiv preprint arXiv:2308.11430, 2024.

[9] T. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020.

8. ACRONYMS

GCIA	Ground Combat Systems Common Infrastructure Architecture
LLM	Large Language Model
MBSE	Model Based Systems Engineering
MOSA	Modular Open Systems Approach
SysML	System Modeling Language