Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

A SIMULATION-ENABLED FRAMEWORK FOR MISSION ENGINEERING PROBLEM DEFINITION: INTEGRATING AI-DRIVEN KNOWLEDGE RETRIEVAL WITH HUMAN-CENTERED DESIGN

Rafi Soule[1], and Barry C. Ezell[2]

[1]Dept. of Eng. Management and Systems Engineering, Old Dominion University, Norfolk, VA, USA
[2]VA Modeling, Analysis, and Simulation Center (VMASC), Old Dominion University, Suffolk, VA, USA

ABSTRACT

Mission Engineering (ME) requires coordination of multiple systems and stakeholders, but often suffers from unclear problem definitions, fragmented knowledge, and limited engagement. This paper proposes a hybrid methodology integrating Retrieval-Augmented Generation (RAG), Human-Centered Design (HCD), and Participatory Design (PD) within a Model-Based Systems Engineering (MBSE) framework. The approach generates context-rich, stakeholder-aligned mission problem statements, as demonstrated in the Spectrum Lab case study, ultimately improving mission effectiveness and stakeholder collaboration.

1 INTRODUCTION

Mission Engineering in the U.S. Department of Defense (DoD) applies systems engineering principles to plan, analyze, and design missions involving multiple systems, focusing on mission outcomes in complex environments. Despite its importance, ME often struggles with unclear problem definitions, stakeholder misalignment, and scattered knowledge. Recent DoD policy emphasizes interdisciplinary, user-inclusive approaches to reduce integration risk. HCD and PD address these gaps by prioritizing end-user needs and collaborative solution development. Our hybrid framework combines RAG, HCD, and PD within an MBSE process, ensuring problem definitions are data-driven and stakeholder-validated.

Human-Centered Design principles are employed to keep the process focused on the people and stakeholders for whom the mission is being engineered. HCD emphasizes empathy with users (or in this case, warfighters, operators, and other mission stakeholders), iterative prototyping, and feedback loops. In practical terms, we incorporate HCD by gathering user needs and context through interviews, surveys, and workshops at the outset of the project. Participatory Design goes a step further by actively involving those stakeholders in co-creating the solution. Rather than engineers working in isolation, PD methods bring in stakeholders (e.g. mission planners, analysts, end-users) to collaborate in formulating the mission problem and to validate the relevance of retrieved information and AI-generated suggestions. This participatory approach ensures that the problem definition resonates with real operational concerns and garners buy-in from the community. It also helps surface tacit knowledge that might not be documented elsewhere. By embedding HCD and PD into the MBSE workflow, the framework ensures that human insights drive the modeling effort rather than being an afterthought.

We propose a hybrid framework that integrates emerging AI techniques with human-centered methodologies to address this shortcoming. The framework combines RAG, HCD, and PD within an MBSE-driven process. RAG is an approach from the AI domain where a large language model is paired with a knowledge retrieval mechanism to generate outputs grounded in real data (Lewis et al., 2020). In our context, RAG allows the engineering team to tap into extensive knowledge repositories – such as organizational lessons learned, prior mission reports, and even public databases of mission statements or requirements – to inform the mission definition. By automatically retrieving relevant documents and facts, the AI can draft preliminary mission problem statements or key requirements that are backed by evidence.

This helps avoid the "starting from scratch" problem and reduces the risk of LLM hallucinations by ensuring generated content is traceable to sources.

2 BACKGROUND AND RELATED WORK

2.1 Mission Engineering and Problem Framing

The Department of Defense (DoD) and systems engineering community increasingly recognize the importance of Mission Engineering (ME) – treating missions as adaptable, designable systems (Beery & Paulo, 2019). A fundamental early step in ME is problem framing: identifying mission needs, objectives, and constraints before jumping to solutions. Poorly framed mission problems lead to suboptimal designs or failure to meet true operational needs. Prior work highlights that mission success depends on aligning technical systems with operational context and user needs. Thus, methodologies that can rigorously capture both *technical and human dimensions* of a mission problem are needed. Our work addresses this by fusing techniques from AI (for knowledge retrieval) and human-centered methods into a modeling framework.

ME is an interdisciplinary process for analyzing, designing, and integrating operational capabilities to achieve mission goals (Dahmann & Parasidis, 2024), encompassing activities such as mission capability and trade-off analyses as well as interoperability assessment (Giachetti & Hernandez, 2024), aimed at creating mission-oriented systems of systems (SoS) rapidly deployable to address threats or opportunities. As an emerging discipline, ME integrates engineering and analytical domains to develop mission-level capabilities. The DoD's Office of Systems Engineering & Architecture underscores interdisciplinary approaches in ME, highlighting risk management, system integration, and SoS architectures to reduce integration risk (Dahmann et al., 2019). This interdisciplinary focus aligns with human-centered design (HCD) and participatory design (PD) principles: HCD emphasizes empathy and PD involves diverse stakeholders, supporting ME's collaborative nature. Modern ME approaches are increasingly structured and data-driven, emphasizing mission-based inputs (requirements, prototypes, design options) to guide system development to meet operational needs (Goldenberg, 2022). Integrating HCD and PD can strengthen these approaches by making requirements development more inclusive; involving end-users early ensures systems are technically sound and meet workflow expectations (Goldenberg, 2022). Hutchison et al. (2018) identify a framework of ME competencies (governance, operational concepts, interpersonal skills, and leadership) that overlap with human-centered approaches. Emphasis on interpersonal skills and leadership in ME competencies mirrors PD's collaborative nature, and focus on operational concepts aligns with HCD's focus on user context (Hutchison et al., 2018; Vesonder et al., 2017). These insights suggest that integrating HCD and PD into ME could bolster ME practices involving complex human–system integration and multi-stakeholder coordination.

2.2 Retrieval-Augmented Generation (RAG)

RAG systems combine pre-trained language models with external knowledge bases, enabling context-aware information retrieval and synthesis. In simulation contexts, RAG can automatically identify relevant domain knowledge, enhancing transparency and traceability (Lewis et al., 2020).

2.3 Human-Centered Design (HCD)

HCD is a design philosophy and process that keeps end-user needs and usability at the forefront. In a mission context, HCD means engaging with operators, analysts, and other humans in the loop to ensure the problem framing and eventual solutions reflect real-world workflows, limitations, and preferences. HCD emphasizes empathy with users, iterative refinement, and a holistic view of user experience. Prior studies in defense and emergency planning show that failing to account for human factors (cognitive load, training, trust in systems) can derail mission outcomes (Norman, 2014; Shattuck, 2017). We incorporate HCD by involving end-users early: validating the AI-generated context against on-the-ground reality, uncovering unarticulated needs, and iterating on the problem statement with user feedback. This ensures *what* the mission is trying to achieve and *why* is informed by those who ultimately execute the mission.

2.4 Participatory Design (PD)

Participatory Design extends HCD by actively involving stakeholders in co-design. In mission engineering, PD manifests as workshops or collaborative sessions where operators, commanders, engineers – even adversarial experts – jointly define the mission problem and requirements. This democratizes the process: stakeholders don't just inform design; they share authority in shaping it. Prior work has shown PD can increase buy-in and surface diverse perspectives, which is vital in complex missions with many stakeholders (Schuler & Namioka, 1993; Smith et al., 2020). Our framework leverages PD after an initial problem draft is available, to let stakeholders critique and modify it live. The result is a consensus-built problem definition that resonates with all parties, reducing later misunderstandings.

2.5 Model-Based Systems Engineering (MBSE) in Mission Context

MBSE provides a formalized approach to modeling complex systems and their interactions, using languages like SysML (Systems Modeling Language). In mission engineering, MBSE offers a way to create *digital mission models* – capturing not only system architecture but also operational processes and scenarios (Gemma et al., 2022). For instance, SysML activity diagrams can represent mission workflows, and state machine diagrams can model system behavior in different operational modes. Friedenthal *et al.* (2011) demonstrated how SysML can rigorously represent system requirements, behavior, and structure, improving consistency and traceability in system design (Friedenthal et al., 2011). In the mission context, MBSE allows engineers to *simulate performance under various conditions* and evaluate different system-of-systems configurations virtually. In fact, Digital Mission Engineering (DME) is an emerging paradigm defined as *"using digital modeling, simulation, and analysis to incorporate the operational environment and evaluate mission outcomes"* (Ansys, 2020). Our work aligns with this: by formalizing the mission problem in a SysML model, we set the stage for simulation-based analysis of that mission. Integrating HCD into MBSE means including human task models and user workflows in these mission models. For example, one can model how an operator interacts with a system in a mission thread alongside system functions (Gemma et al., 2022). This helps identify usability bottlenecks or training gaps early, before any real deployment. Participatory modeling (an aspect of PD in MBSE) further means stakeholders help build or review SysML diagrams (e.g., mission use-case diagrams or operational views), ensuring the model reflects operational reality. Literature has begun to fuse MBSE with simulation for mission analysis; for example, Beery and Paulo (2019) map SysML architectural products to an analysis approach that integrates operational simulations (Beery & Paulo, 2019). Batarseh and McGinnis (2012) similarly showed that SysML models can be transformed into discrete-event simulation models (Arena) to assess system performance quantitatively (Batarseh & McGinnis, 2012; McGinnis & Ustun, 2009). These efforts underscore that MBSE and simulation are complementary: MBSE provides the authoritative model, and simulation uses that model to generate dynamic insights. We leverage this synergy by treating the MBSE model not just as documentation but as an *executable mission prototype*.

2.6 Summary of Gaps

The convergence of AI-driven knowledge retrieval, human-centered methods, and MBSE is still emerging within ME. Existing ME practices lack a systematic way to incorporate vast prior knowledge (which RAG can provide) and often treat human factors as secondary. Likewise, while MBSE and simulation are recognized in ME, they typically come into play after requirements are set – not during problem formulation. Our framework explicitly fills these gaps by bringing RAG, HCD, PD into the *problem framing* stage and tightly coupling the outcome with simulation-capable MBSE models. This foundation aims to ensure that when the engineering of solutions begins, it is built on a well-vetted, stakeholder-endorsed, and analytically sound problem definition.

Integrating RAG, HCD, and PD in the framing phase makes the process data-rich and user-centric. RAG mechanisms retrieve relevant domain knowledge from archives and literature, broadening the evidence base and reducing oversight. Human-centered and participatory design ensure stakeholder needs, experiential insights, and organizational constraints shape the requirements from the outset (Boy, 2018). Embedding these enriched insights in MBSE models allows teams to simulate and refine mission scenarios during problem definition, tightening the feedback loop between formal modeling and stakeholder feedback (Barclay, 2025; Boy, 2018). This combined approach produces an early problem statement that is both analytically rigorous and collaboratively validated, giving practitioners confidence that solution development will proceed from a well-understood and broadly vetted foundation.

3 PROPOSED METHODOLOGY

To address the identified gaps, we propose a Human-Centered Problem Definition framework for mission engineering that integrates RAG, HCD, and PD within an MBSE-centric process. The goal is to enhance the problem definition phase of mission engineering by automatically grounding it in relevant knowledge and ensuring it is iteratively refined with stakeholder input. Figure 1 provides an overview of the RAG concept, and Figure 3 (later in this section) is organized into four main stages, with simulation modeling serving as a critical follow-on for validation. Below, we describe each stage and highlight how SysML-based simulations become relevant once the problem definition is in place.

3.1 AI-Driven Knowledge Retrieval with RAG

RAG is an approach from the artificial intelligence domain that can retrieve and utilize domain knowledge during content generation. Instead of relying solely on a pre-trained model, a RAG system first searches a knowledge base for relevant information (documents, databases, prior cases) and then uses that information to generate a context-specific response or summary. In the context of mission engineering, a RAG system can be used to automatically gather information related to a given mission problem statement or scenario, providing a starting point for problem definition that is grounded in existing data (doctrine, previous mission after-action reports, technical specifications, etc.).

Figure 1: RAG pipeline consists of four stages: (1) Indexing – domain knowledge (documents, databases, lessons learned) is indexed for search; (2) Retrieval – given a query or mission context, relevant information is retrieved from the index; (3) Augmentation – the retrieved information is compiled and fed into an AI reasoning process; and (4) Generation – the AI generates a response (e.g., a draft mission problem statement or situation analysis) using both its trained knowledge and the retrieved context. By incorporating retrieval, RAG ensures that the output remains grounded in real data, enabling domain-aware problem framing with transparent sourcing of information. In essence, RAG can provide an initial mission context overview or problem definition that is backed by evidence (e.g., citing relevant doctrine or historical mission data), which can be invaluable for mission engineers at the start of the planning process.

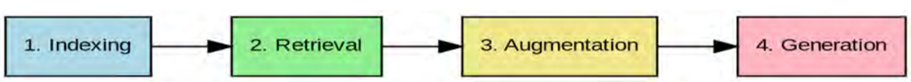

Figure 1: RAG pipeline consists of four stages.

In our methodology, the RAG component serves as a tool for real-time contextual understanding during problem formulation. For example, if the mission under consideration is a humanitarian relief operation, the RAG system might retrieve information about similar past operations, known logistical challenges, environmental factors, and stakeholder roles. The generated output could be a first draft of the mission's problem statement or a list of key considerations, complete with references to the source documents (providing traceability). This helps ensure that the problem framing is informed by a broad base of knowledge rather than limited to the immediate team's memory or perspective. It addresses the "knowledge integration" issue by automating access to relevant information and presenting it in a usable form to the engineering team.

3.2 Integrating HCD and PD Principles

While RAG brings in data and domain knowledge, Human-Centered Design and Participatory Design ensure that the process remains focused on the people involved in the mission. We incorporate HCD and PD in the methodology through structured stakeholder engagement and iterative refinement steps:

- Human-Centered Design in Problem Definition: After the RAG system produces an initial problem context, mission engineers apply HCD practices to interpret and refine that information. This involves validating that the identified problems are the "right" problems from a user perspective. For instance, the team would conduct interviews or empathy mapping with end-users (e.g., field operators, decision-makers) to understand their pain points and priorities relative to the mission context. HCD encourages the team to question assumptions: Are the mission goals and constraints framed in a way that reflects on-the-ground reality? What unarticulated needs might the operators have? The RAG output serves as a starting hypothesis that the HCD process then critiques and elaborates on, ensuring human concerns (usability, workload, communication, etc.) are brought to light.
- Participatory Design Workshops: We explicitly include PD by holding co-design sessions (workshops) with stakeholders after the initial problem statement is drafted. In these sessions, stakeholders – such as operational commanders, system operators, analysts, and possibly adversarial perspective experts – are invited to collaboratively review and modify the mission problem definition. They have access to the RAG-provided information and any initial models or diagrams. Using participatory methods (brainstorming, storyboarding mission scenarios, voting on mission priorities), the group can identify errors or omissions in the problem framing and contribute their knowledge. This process ensures shared decision-making in defining the mission scope and objectives, which builds buy-in and surfaces diverse perspectives early. By the end of PD workshops, the problem statement and associated requirements are not just handed down by engineers; they are co-created, which significantly improves stakeholder alignment.

HCD focuses on understanding users through empathy, iteration, openness, and inclusion of diverse perspectives; PD emphasizes co-design, shared decision authority, and real-time feedback from users. Both approaches overlap in aiming for stakeholder-aligned problem formulation, as depicted by the convergence toward a shared understanding of the mission problem. In practice, this means the mission problem is defined not only by technical requirements but also by the insights of operators and stakeholders, leading to a formulation that all parties understand and support. By combining HCD and PD, the methodology ensures that mission problem definition is both user-informed (via HCD's user research and iterative refinement) and user-influenced (via PD's direct stakeholder participation). This overlap results in greater consensus on what the mission is solving and why, reducing the risk of misalignment between what is engineered and what is actually needed in the field.

3.3 Human-Centered Problem Definition Framework

The overall process framework, which brings together RAG, HCD, and PD within an ME context, is structured in sequential stages (with some iteration within and between stages as needed). This framework ensures the mission problem is well-defined, human-vetted, and ready for model-based simulation and trade studies. Figure 2 illustrates the stages of the proposed Human-Centered Problem Definition framework for an MBSE-driven Mission Engineering process:

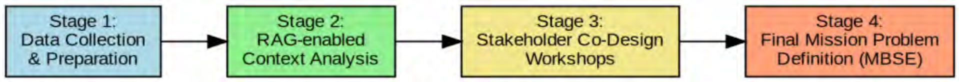

Figure 2: Proposed Framework for integrating RAG with HCD/PD in ME.

- **Stage 1 – Data Collection & Preparation:** The process begins by gathering and organizing mission-relevant data. This includes doctrine documents, prior mission after-action reports, stakeholder interviews, technical specifications, etc. All these sources populate a knowledge repository that the RAG engine will index. Concurrently, initial stakeholder inputs on mission goals, pain points, and assumptions are captured. By building this repository, we ensure the AI has high-quality, context-rich data to draw from, and we document any initial hypotheses or constraints the stakeholders have (e.g. "Mission must be completed within 48 hours" or "Communications infrastructure is unreliable in the area"). This stage lays the groundwork for an AI-informed analysis by structuring raw knowledge into a usable form.
- **Stage 2 – RAG-Enabled Context Analysis:** In this stage, the RAG engine is run against the Stage 1 repository to generate a draft mission context and problem statement. The RAG system, given a query or prompt about the mission (for example, *"Outline the main challenges and objectives for a humanitarian relief mission in a coastal region after a hurricane"*), retrieves relevant facts and lessons from the repository. It then produces a synthesized output such as a situation overview, key challenges, objectives, and even suggested requirements, complete with references to the source data. The result might be a few paragraphs describing the mission environment (terrain, weather, adversarial presence if any), known logistical or operational challenges (e.g. port infrastructure damage, need for disease control), and objectives (rescue and supply distribution priorities), each traceable to real past instances or doctrine. This draft problem statement serves as a starting point – grounded in evidence – for the human team. It's important to note the RAG output is not final or authoritative; rather, it jump-starts the human-centered refinement by ensuring the team's initial focus is comprehensive and data-driven. It mitigates cognitive biases and knowledge blind spots by injecting outside information. In short, Stage 2 yields a proto-problem-definition informed by AI.
- **Stage 3 – Stakeholder Co-Design Workshops:** Next, we put the AI-generated draft into the hands of human stakeholders through participatory design workshops. In facilitated sessions, mission stakeholders (such as commanders, field operators, analysts, domain experts – including red-team or adversary perspectives) collaboratively review the draft problem statement. They discuss and critique the content: *Are the stated mission goals correct? What's missing? Are certain challenges overstated or understated?* Using HCD techniques, the workshop encourages empathy and uncovering real needs: participants might share on-the-ground anecdotes or concerns not captured in documents. We incorporate interactive tools in these workshops – for example, a shared interface (our prototype *Spectrum Lab*, discussed later) where participants can edit the text, comment, or vote on priorities in real time. Visual aids generated from the MBSE model (which is evolving in parallel) can be used to facilitate understanding – e.g. an automatically generated mission thread diagram or an operational node connectivity chart that stakeholders annotate. Through these iterations, the problem statement is *refined and corrected*. PD ensures that *diverse viewpoints are integrated:* in one use-case, we found that including an electronic warfare officer and a communications officer in a scenario planning workshop highlighted a conflict between jamming threats and comms needs, leading to a new requirement about resilient communications. By the end of Stage 3, we expect to have a significantly improved problem definition: one that is not only data-driven (from Stage 2) but also *experience-driven* from practitioners. The output of this stage is a revised mission problem statement and a preliminary set of mission requirements or key considerations, which have been co-created and agreed upon by the group. MBSE modeling is happening concurrently as stakeholders identify mission threads or use cases, systems engineers can start capturing those in SysML diagrams (use case diagrams, activity flows, etc.). These diagrams or mission thread models might be iteratively updated during the workshop to reflect

the evolving understanding. This dynamic interplay keeps the digital model in lockstep with stakeholder consensus.

- **Stage 4 – Final Mission Problem Definition:** After the workshops, the team synthesizes all the inputs into a formal mission problem definition. This includes a finalized problem statement narrative and a structured set of mission requirements, constraints, and assumptions. Uniquely, this final output is captured within an MBSE model (or linked document repository) as the single source of truth. In practice, by Stage 4 the SysML model contains an authoritative representation of the mission problem space. For example, a SysML requirements diagram now lists the mission's objectives and key requirements (both functional and non-functional) and traces them to stakeholders or source data. An operational concept diagram (OV-1 or use case diagram) depicts the high-level scenario, and maybe activity diagrams outline the mission workflow. All of these were iteratively refined in Stage 3 and now baselined in Stage 4. Formalizing the problem in MBSE ensures consistency and prepares for downstream engineering – the transition from problem definition to solution design/analysis is now seamless. The final problem definition is richly informed by real data and vetted by stakeholders, providing a solid foundation for the rest of the mission engineering lifecycle.

Role of Simulation Modeling (Post-Stage 4): Although our framework's core problem-framing stages are as above, we explicitly integrate simulation modeling immediately after Stage 4 as a bridge to analysis. At this point, we have an MBSE model of the mission – and we leverage it to conduct SysML-based behavioral simulations that support or validate the problem framing. In other words, **Stage 4a (Simulation-Based Validation)** uses the MBSE artifacts as input to simulations before committing to detailed design. Many MBSE environments support executing SysML models or connecting them with simulation tools. For instance, MagicDraw (Cameo) with the Cameo Simulation Toolkit can execute SysML state machine and activity diagrams directly. Parametric constraint diagrams can be solved to compute performance measures, or even co-simulated with external physics models (Spangelo et al, 2013). We exploit these capabilities to simulate the *mission scenario* defined by the problem statement. For example, if the mission involves a series of tasks (deliver supplies, survey area, evacuate wounded) captured in an activity diagram, we can run that diagram as a discrete-event simulation to estimate how long the mission might take, where bottlenecks occur, or how resource usage (fuel, supplies) accumulates over time. Similarly, if our problem definition includes a requirement like "maintain 90% communication uptime under jamming," we can use a state machine model of the communications network to simulate uptime vs jamming conditions and see if 90% is achievable with the assumed constraints. This kind of behavioral simulation allows us to verify that the problem statement is realistic and internally consistent. It answers questions like: *Given the defined mission tasks and constraints, are the objectives achievable? Which requirements are most sensitive or risky?* By performing simulation at the end of problem framing, we effectively do a "sanity check" using quantitative analysis before moving to design solutions. This approach reflects best practices in simulation-based methodology – using models early for validation. It also aligns with the concept of *digital mission engineering*, where the digital model of the mission is continuously used for analysis (Beery & Paulo, 2019). Any insights from these simulations feed back into our framework: if a simulation run uncovers an unanticipated issue (e.g., a particular step consistently takes too long or a resource is overutilized), we treat that as a new data point. Stakeholders and the RAG engine can incorporate this finding – perhaps by adding a new constraint or revising a requirement – and we adjust the problem definition accordingly. This closes the loop, ensuring our defined problem is not only stakeholder-approved but also simulation-tested. In summary, *simulation modeling supports our framework by extending Stage 4 into an analysis phase*: the MBSE model (output of Stage 4) becomes an executable simulation model. In this case, the problem definition doesn't remain a static document but becomes a scenario that can be played out and studied through modeling & simulation. Figure 2 in our case study (next section) illustrates this by showing an "MBSE Model & Simulation" component as part of the tool architecture, indicating how the finalized problem definition feeds into simulation and feeds back results. We emphasize that while we present simulation as a post-stage activity, it is an integral part of the methodology's spirit. *The inclusion of simulation ensures that the human-centered, AI-informed problem framing is grounded in quantitative reality.* It provides confidence that the mission problem, as defined, has been stress-tested and understood dynamically, which ultimately de-risks the downstream engineering of solutions.

3.4 Spectrum Lab Case Study: Prototype Implementation

To demonstrate the methodology, we developed a prototype tool called *Spectrum Lab* and applied it to a notional mission scenario. Spectrum Lab serves as an interactive platform implementing the framework's stages: it incorporates an AI RAG engine, a collaborative stakeholder interface, and an MBSE model repository with simulation capabilities. The scenario we use for illustration is a mission engineering problem involving electronic warfare and communications in a contested environment (e.g., ensuring resilient radio communications for a platoon under jamming during an operation).

Spectrum Lab Workflow: In Stage 1, we populated Spectrum Lab's knowledge base with documents about spectrum management, electronic warfare tactics, previous exercises, and relevant communication system specs. Stakeholders initially indicated a goal to "maintain comms for platoon leaders under enemy jamming" and concerns about spectrum congestion. In Stage 2, the RAG engine (built on GPT-4 with retrieval plugins) indexed this corpus. Given a prompt about *"mission context for maintaining communications under jamming"*, it produced a draft summary: outlining adversary jamming capabilities, environmental factors (terrain blocking line-of-sight), possible mitigation strategies (frequency hopping, relay drones), and objectives like a 90% uptime requirement for leader comms. This draft cited a NATO electronic warfare report and a past training exercise where communication loss was a problem.

In Stage 3, we convened a workshop with three roles simulated by our team: an electronic warfare officer, a communications officer, and a mission commander (plus systems engineers facilitating). Using Spectrum Lab's interface, the team reviewed the AI draft. The *comms officer* noted the draft didn't consider encryption key distribution (which affects comms setup time) – a gap we added as a new consideration. The *EW officer* stressed that the 90% uptime requirement might be too low given modern operations; they suggested aiming for 99% and identified a critical scenario when all platoon leaders transmit simultaneously. The *commander* prioritized ensuring any solution is simple for soldiers to use (a human-factor point). As they discussed, the MBSE model (in Cameo) was updated live: a use case diagram for "Ensure Platoon Comms" was modified to include an alternate flow for rekeying encryption, and an activity diagram was adjusted with an explicit jamming mitigation step. Spectrum Lab's interface allowed the stakeholders to see these diagrams. They collectively edited the problem statement text – increasing the uptime requirement to 95% as a compromise, and adding a constraint about limited training time for soldiers to learn new equipment. By the workshop's end, the mission problem definition was much more complete and agreed-upon: essentially, *"How can we ensure at least 95% radio communication uptime for platoon leaders in a 5 km² area under continuous jamming, using equipment and procedures simple enough for a single soldier to operate, within 48 hours of mission start?"* along with about a dozen key requirements/constraints. This was stored in the SysML model (requirements diagram and accompanying documentation).

In Stage 4, Spectrum Lab automatically compiled the final problem statement and requirements into the MBSE repository (Cameo model). We now had a digital artifact representing the mission problem. At this point, we invoked the *simulation extension*: using a simplistic *spectrum allocation* simulation we built. The simulation took the communication network described in the MBSE model (nodes for platoon leaders, a base station, an adversary jammer with certain range) and ran a time-step model of communications and jamming. We ran a quick simulation experiment varying the number of frequencies and power of the jammer. The result showed that with the current assumptions, ~92–98% uptime could be achieved – confirming that the 95% requirement was reasonable, but also revealing that if the jammer were even 10% more powerful, uptime would drop below 90%. This insight was fed back into the problem definition: the stakeholders decided to add a note about jammer signal strength assumptions (i.e., the problem definition now explicitly states the assumed jammer capabilities – a refinement prompted by simulation). This small case study highlights the value of integrating simulation at the problem stage: if the simulation had shown, say, that 95% was utterly unattainable with any plausible setup, the requirement could be reconsidered immediately. Instead of discovering this flaw during design or testing, it was caught on the whiteboard, so to speak.

Throughout this process, the Spectrum Lab tool integrates technology and people. Figure 4 shows its architecture, highlighting connections between the RAG engine, stakeholder interface, and MBSE model repository.

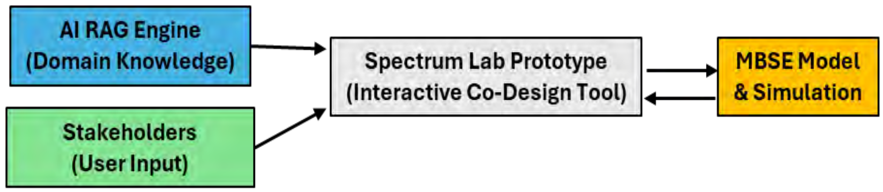

Figure 3: Spectrum Lab Prototype Architecture.

The Spectrum Lab tool serves as an interactive co-design platform connecting three key components: (1) an *AI RAG Engine* linked to a mission knowledge base (doctrine, prior cases, technical data), which provides context and suggestions; (2) the *Stakeholder Interface*, where mission stakeholders (operators, analysts, decision-makers) input their knowledge, review AI-generated content, and collaboratively refine the mission problem definition; and (3) the *MBSE Model & Simulation Environment*, which receives the finalized problem definition and provides means to validate and explore it (e.g., running simulations or visualizations to ensure the problem is well-understood and to inform solution trade-offs). The architecture allows information to flow in both directions: the RAG engine supplies data to stakeholders; stakeholders' inputs update the MBSE model; and simulation results or model constraints feedback to refine understanding. This closed-loop ensures consistency – if a simulation reveals a new challenge (e.g., a particular frequency band is too congested), that insight is sent back into the RAG knowledge base or noted by stakeholders, and the problem definition is adjusted accordingly. In our case study, this architecture enabled rapid iteration: the AI provided a draft, humans adjusted it, and the model/simulation checked it, all in one integrated workflow.

3.5 Discussion

The demonstration, though preliminary, illustrates the *practical benefits* of our integrated framework. First, by combining RAG with HCD/PD, we observed improved *problem comprehension* and stakeholder buy-in. Users of the prototype commented that the AI-generated context surfaced non-obvious information (e.g., an old lesson learned that the team had overlooked), and the collaborative interface made them feel actively engaged in defining the mission (addressing the common issue of stakeholders feeling left out early on). These qualitative findings support our claim that front-loading the engineering process with AI-informed and human-centered activities can reduce knowledge gaps and misalignment. Second, the immediate formalization of the problem in an MBSE model created a **digital thread** from problem definition to solution development. This is aligned with the concept of a single source of truth in digital engineering – any updates in the problem definition automatically reflected in the model, and vice versa.

The integration of simulation modeling into the framework proved valuable. By running a quick SysML-driven simulation on the problem definition, the team received early feedback on the feasibility of requirements. This kind of simulation-before-design is not standard in many mission engineering practices, yet our results suggest it can catch issues that would otherwise appear much later. Essentially, we treated the problem statement itself as a hypothesis to be tested via modeling & simulation. This approach is consistent with model-driven engineering principles and *digital mission engineering* as defined by industry (Ansys, 2020; Gemma et al., 2022) – we are executing the "mission" in a virtual environment to validate that we're solving the right problem. The literature on MBSE in early life-cycle supports this notion: by exploring mission threads through simulation, one can perform trades and sensitivity analyses on requirements before locking them down (Beery & Paulo, 2019). For example, Had we found that the 95% comm uptime was consistently unachievable, the stakeholders could have revisited that requirement or introduced new mitigating measures *before* any system design began. This reduces costly rework and helps ensure that subsequent engineering efforts are spent on viable solutions. Additionally, integrating simulation aligns our framework with the – it demonstrates how architecture modeling languages like SysML and simulation tools can work in concert. Others have noted challenges in bridging MBSE models to simulation (e.g., ensuring that SysML conceptual models correctly translate to executable models) (McGinnis & Ustun, 2009). Our approach offers a template for this bridge: by constraining the MBSE model to a well-defined mission scenario and using off-the-shelf simulation integration (Cameo's built-in execution and a custom domain-specific simulator), we achieved a fluid workflow. In doing so, we contribute to the ongoing dialogue on how to make *simulation a first-class citizen in MBSE* – not an afterthought.

3.6 Challenges and Considerations

There are, of course, limitations and areas for further development. The prototype was developed in a controlled lab setting within a limited domain (spectrum management). Scaling this approach to real-world complexity – such as joint operations involving numerous systems and stakeholders – presents major challenges. The following sections outline these limitations and potential areas for advancement.

There are limitations and areas for further development. The prototype was developed in a controlled lab setting within a limited domain (spectrum management). Scaling this approach to real-world complexity, such as joint operations involving numerous systems and stakeholders, presents major challenges. The following sections outline these limitations and potential areas for advancement.

- Limited Domain & Generalizability: The current prototype was a controlled lab scenario focused on spectrum management. Generalizing and scaling this methodology to real-world mission engineering contexts (e.g., joint operations with dozens of systems and diverse stakeholders) will pose challenges. The approach may need adaptation for unique or complex scenarios, and further research is needed to validate it across a wider range of ME problems.
- Dependence on High-Quality Data (RAG): The RAG component is only as effective as the data it draws on. If the knowledge base is incomplete, outdated, or biased, the AI may produce inaccurate or irrelevant outputs. Continuous curation of high-quality, relevant data is essential to avoid misinformation. There is also a risk of overwhelming stakeholders with too much retrieved information, so the system must filter and present data judiciously.
- Simulation Fidelity & Integration: Ensuring realistic simulations is another concern – our prototype's quick simulation was simplistic and may not capture complex real-world behaviors. In larger settings, integrating more sophisticated models would be necessary (for example, linking the SysML design to high-fidelity discrete-event simulations or physics-based tools like Ansys STK for orbital and communication analysis). Fortunately, our MBSE approach is compatible with such extensions; tools like ModelCenter can orchestrate complex simulation workflows directly from SysML models (Spangelo et al, 2013). This compatibility will support scaling up the fidelity of analyses as needed.
- User Adoption & Training: Cultural adoption is an issue, since many mission engineers and stakeholders are unfamiliar with AI-driven tools or SysML models. Our framework represents a shift in early mission planning, which may require dedicated training and new roles (e.g., a facilitator to operate a Spectrum Lab–like tool). Users may be hesitant to trust an AI assistant initially, perceiving it as a "black box."
- Trust & Traceability: Early user feedback showed initial skepticism about the AI's recommendations. We found that making the AI's outputs transparent – clearly showing the source of each

recommendation – helped build confidence. Emphasizing traceability of RAG outputs (so users can verify information back to its source) and providing a user-friendly interface are key for real-world acceptance.

- Success Criteria for Evaluation: Finally, evaluating the effectiveness of this methodology requires clear success criteria. In our prototype, we gauged success by stakeholder satisfaction with the problem-definition process, the time required to reach consensus on decisions, and the accuracy of the AI-assisted outputs against expected results. Defining and measuring these metrics is itself a challenge, but it is crucial for demonstrating the approach's value and guiding future improvements.

4 CONCLUSION

We presented an integrated framework that strengthens the mission problem framing stage by coupling AI-driven knowledge RAG with human-centered design principles, all anchored in an MBSE process that feeds directly into simulation modeling. This approach ensures that mission engineering begins with a well-informed, stakeholder-validated problem definition that can be analyzed quantitatively before solution development. By doing so, it improves the odds that subsequent engineering efforts address the right problem and meet operational needs. The incorporation of SysML-based simulation in particular extends the framework's impact beyond documentation – it provides a methodological connection to simulation modeling. Our case study, while limited in scope, demonstrated the potential for rapid feedback and iterative refinement when these elements work in unison.

Moving forward, we plan to apply this framework to more complex case studies (e.g., multi-domain operations) and measure its benefits in terms of reduced rework, improved stakeholder satisfaction, and mission outcome success. We will also explore integrating more advanced simulation tools and AI reasoners (for example, using RAG not just for text generation but to interface with simulation results, closing the loop with automated suggestions). Ultimately, we see this work as a step toward *digital-first mission engineering*, where problem formulation, design, and analysis are all part of a connected, model-driven continuum. We hope that this approach inspires further research at the intersection of MBSE, AI, and simulation in the mission engineering community, and we welcome collaboration and critique from practitioners and researchers in these domains.

ACKNOWLEDGMENTS

This work has been supported by the Spectrum and Advanced Training Technology Laboratory at Old Dominion University. The authors gratefully acknowledge the contributions of the research and engineering teams involved in the development and integration of mission engineering frameworks.

REFERENCES

Ansys. (2020). *AGI: Digital Mission Engineering Introduction (1 of 2)*. https://www.agi.com/resources/videos/dme-intro-part-1, accessed 7th June 2025.

Barclay, R. B. (2025). *INCOSE: Generative AI for Systems Engineering: Advances, Reasoning Models, and Deployment Paradigms | LinkedIn / INCOSE*. https://www.linkedin.com/pulse/generative-ai-systems-engineering-advances-reasoning-models-deployment-df8zc/ , accessed 14th June 2025.

Batarseh & McGinnis. (2012). *SysML to Discrete-event Simulation to Analyze Electronic Assembly Systems*. https://www.researchgate.net/publication/262166285_SysML_to_discrete-event_simulation_to_analyze_electronic_assembly_systems , accessed 7th June 2025.

Beery, P., & Paulo, E. (2019). Application of Model-Based Systems Engineering Concepts to Support Mission Engineering. *Systems*, *7*(3), 44. https://doi.org/10.3390/SYSTEMS7030044

Boy, G. A. (2018). Human-centered Design of Upcoming Human Missions to Mars. *Journal of Space Safety Engineering*, *5*(1), 9–13. https://doi.org/10.1016/J.JSSE.2018.01.001

Dahmann & Parasidis. (2024). Mission Engineering. *The ITEA Journal of Test and Evaluation*, *45*(3). https://doi.org/10.61278/ITEA.45.3.1001

Dahmann, Doskey, S., & Tolk, A. (2019). *Mission Engineering, Systems Engineering and Systems of Systems Engineering*. https://mitre.tahoe.appsembler.com/asset-v1:mitre+SOS0+open+type@asset+block@04-21-2020-SoSECIE-Tolk-Mission_Engineering_SE_and_SoSE.pdf , accessed 15th August 2025.

Friedenthal, S., Moore, A., & Steiner, R. (2011). A Practical Guide to SysML: The Systems Modeling Language, Second Edition. *A Practical Guide to SysML: The Systems Modeling Language, Second Edition*, 1–615. https://doi.org/10.1016/C2010-0-66331-0

Gemma, R., Habig, R., & Brechmann, M. (2022). *How Digital Mission Engineering will shape your future defense systems, to increase effectiveness and efficiency. NATO S&T Organization, Milan, Italy, accessed 11th August 2024.*

Giachetti & Hernandez. (2024). *Mission Engineering - SEBoK*. https://sebokwiki.org/wiki/Mission_Engineering , accessed 18th September 2024.

Goldenberg. (2022). *Mission Engineering Methodology Studies and Analysis*. https://ac.cto.mil/wp-content/uploads/2022/03/ME_Methodology_Feb2022.pdf , accessed 19th September 2024.

Hutchison, N., Tao, H. Y. S., Miller, W., Verma, D., & Vesonder, G. (2018). Framework for Mission Engineering Competencies. *INCOSE International Symposium*, *28*(1), 518–531. https://doi.org/10.1002/J.2334-5837.2018.00497.X/FULLPDF

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W. T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, *2020-December*. https://arxiv.org/abs/2005.11401v4

McGinnis, L., & Ustun, V. (2009). A simple example of SysML-driven simulation. *Proceedings - Winter Simulation Conference*, 1703–1710. https://doi.org/10.1109/WSC.2009.5429169

Norman, D. (2014). *The Design of Everyday Things*. https://mitpress.mit.edu/9780262525671/the-design-of-everyday-things/ , accessed 28th March 2025.

Schuler & Namioka. (1993). *Participatory design: Principles and practices.* https://psycnet.apa.org/record/1993-97696-000 , accessed 28th March 2025.

Shattuck, L. G. (2017). From Cognitive Systems Engineering to Human Systems Integration. *Cognitive Systems Engineering*, 309–326. https://doi.org/10.1201/9781315572529-17

Smith, S., Winkler, S., Towne, S., & Lutz, B. (2020). Utilizing CBPR Charrette in Community-Academic Research Partnerships – What Stakeholders Should Know. *Journal of Participatory Research Methods*, *1*(1), 2020. https://doi.org/10.35844/001C.13179

Spangelo et al. (2013). *Model Based Systems Engineering (MBSE) Applied to Radio Aurora Explorer (RAX) CubeSat Mission Operational Scenarios*. IEEE Aerospace Conference. https://doi.org/10.1109/AERO.2013.6496894.

Vesonder, G., Verma, D., Hutchinson, N., Miller, M. W., & Yan, H. (2017). *RT-171 Mission Engineering Competencies Sponsor: DASD(SE). 9th Annual SERC Sponsor Research Review, November 8th, 2017, Washington, DC.* www.sercuarc.org

AUTHOR BIOGRAPHIES

RAFI SOULE is a PhD Candidate in Systems Engineering at Old Dominion University. Her research focuses on integrating AI technologies with human-centered methodologies to enhance Mission Engineering practices. She has contributed to multiple research projects involving MBSE and AI in complex mission environments. Her email address is rsoul003@odu.edu.

BARRY C. EZELL, PhD is a research professor at the Virginia Modeling , Analysis, and Simulation Center (VMASC), at Old Dominion University. He specializes in modeling and simulation He has collaborated with state and federal agencies. His email address is bezell@odu.edu.