Title: AI-Enhanced Requirements Traceability Using MBSE and Large Language Models for Complex Systems

Research Area: AI to manage large-scale data-intensive tasks, including mission model creation, tradespace exploration, document review and synthesis (AI4SE Track)

Submission Type: Research Presentation

Primary Author/Presenter: Henock Legesse, Model Based Systems Engineer, NASA

Extended Abstract

1. Introduction and Objectives

Requirements traceability represents one of the most fundamental yet challenging aspects of systems engineering practice [5]. The ability to maintain clear parent-child linkages across requirement hierarchies is essential for ensuring system completeness, managing change impacts, and demonstrating compliance throughout the system lifecycle. However, maintaining proper traceability becomes increasingly difficult as system complexity grows and project constraints intensify.

This research addresses the critical challenge of identifying and correcting poorly linked child requirements in complex systems engineering projects. The primary objectives of this work are to: (1) develop an automated approach for identifying orphaned or incorrectly linked requirements using artificial intelligence techniques [1], (2) reduce the substantial human effort traditionally required for comprehensive requirements traceability analysis, (3) enable early detection of traceability gaps when correction costs are minimal rather than during late-phase reviews [6], and (4) demonstrate how AI can augment traditional systems engineering processes while maintaining necessary human oversight.

The motivation for this work stems from two prevalent scenarios in systems engineering practice. First, projects often inherit requirements and materials from previous analogous missions, and during tailoring for new missions, some requirements may not be properly purged while newly added requirements may lack proper parent linkages. Second, discipline engineers frequently develop products in an environment where systems engineering support is limited and schedule pressures prioritize technical progress over SE rigor [3]. As a result, it is often the case required traceability and verification matrices are done closer to acceptance review rather than as part of the technical maturation. This practice exposes traceability gaps only during acceptance reviews.

2. Methodology and Technical Approach

The solution integrates Model-Based Systems Engineering (MBSE) principles [2] with advanced Large Language Model (LLM) capabilities to create a multi-layered automated analysis system. The methodology consists of five primary phases: requirement capture and organization, context enrichment, indexing and storage, hierarchical document analysis, and confidence-based recommendation.

The foundation begins with MBSE-based requirement capture, systematically organizing requirements across all hierarchical levels targeted for parent-child linkage analysis [4]. This structured approach ensures comprehensive coverage and maintains the architectural relationships essential for effective traceability analysis. The MBSE environment allows the usage of API-developed tools that automate and seamlessly integrate the process.

Context enrichment represents a critical innovation in this approach. Each requirement is enhanced with contextual attributes generated through LLM analysis of both the requirement statement and its surrounding document content. This process addresses three common challenges in requirements analysis: poorly written requirements that lack clarity, requirements that implicitly rely on information from their document sections, and compressed requirement statements that obscure true intent [1]. The context enrichment process utilizes OpenAI GPT-4o to analyze document structure, surrounding paragraphs, section headers, and cross-references to generate comprehensive contextual descriptions that clarify requirement intent and scope.

The enriched requirements are then stored and indexed using vector-based retrieval systems optimized for LLM processing. This indexing strategy enables efficient retrieval during subsequent analysis phases while maintaining the semantic relationships essential for accurate parent-child matching.

The automated analysis proceeds through three sequential LLM layers, each designed to progressively refine the candidate parent requirements. The first layer analyzes each child requirement against summary descriptions of potential parent documents, identifying which documents are most likely to contain appropriate parent requirements. This document-level filtering significantly reduces the search space for detailed analysis.

The second layer performs detailed analysis within each identified document, recommending the top five parent requirement candidates based on semantic similarity, logical relationships, and requirement scope alignment. This layer leverages the enriched context attributes to perform nuanced matching that considers both explicit requirement statements and implicit contextual information.

The final layer employs confidence assessment techniques to evaluate the likelihood of each recommended parent-child relationship. Only matches achieving "high" confidence levels are forwarded to systems engineers for review, while lower-confidence recommendations are filtered out to reduce false positive rates and focus human attention on the most promising candidates. It is important to state the LLM provides recommendation rationale, as LLMs don't rationalize what they do well, the information is used to assess common patterns in poor recommendations to curate better examples for the LLM instructions.

For each LLM layer except for the context creator, a validation layer is created. In an event of hallucinations, the output is checked to be a valid document name or valid parent identification. Although hallucinations are mitigated by engineering prompt systems instruction and providing adequate number of examples, they happen. The validation layer catches these issues and reinstructs the LLM with the error. If a response fails the validation layer again, the process declares "unable to recommend parent requirement".

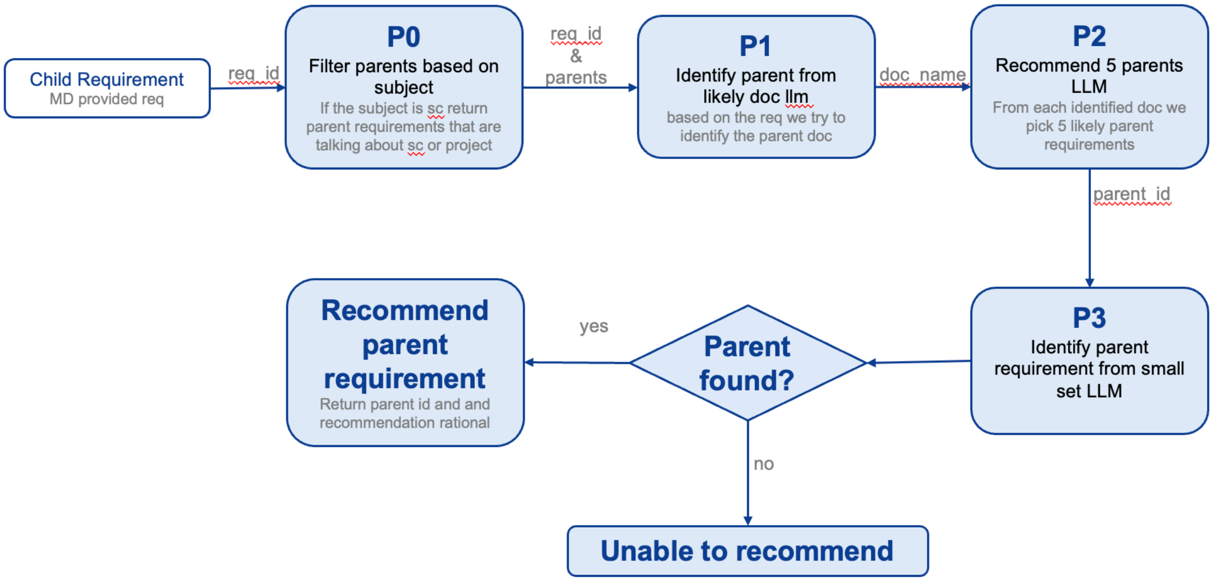

Figure 1: Process flow diagram.

3. Results and Validation

Preliminary validation was conducted on requirement sets representing typical challenging scenarios encountered in systems engineering practice. The test corpus included requirements inherited from previous missions with known traceability gaps, as well as newly developed requirements lacking proper parent linkages.

Across the validation dataset, the system recommended 178 potential parent requirements for analysis. Evaluation by experienced systems engineers rated 55% of these recommendations as strong to moderate matches for identifying appropriate parent requirements. An additional 15% correctly identified the proper parent document but did not pinpoint the exact requirement, indicating partial success that still provides valuable guidance for manual analysis.

These results demonstrate significant capability considering the challenging nature of the test data, which represented poorly developed requirement sets with known traceability issues. Preliminary testing on well-derived requirement sets showed substantially improved performance metrics, suggesting the approach scales effectively with requirement quality and proper SE practice adherence [5].

The time efficiency gains were substantial, with automated analysis completing comprehensive traceability reviews in hours rather than the weeks typically required for manual analysis of similar scope. This efficiency improvement enables more frequent traceability assessments throughout the project lifecycle, supporting early detection and correction of issues.

Additional validation on this solution will be conducted on two other projects and reported in the final paper.

4. Practical Implications and Relevance

This work directly addresses critical pain points in systems engineering practice where traditional manual approaches prove inadequate due to scale, complexity, or resource constraints [6]. The ability to automatically identify traceability gaps enables proactive requirement management rather than reactive problem-solving during critical project phases.

The economic implications are significant, as discovering untraced requirements late in project lifecycles can result in substantial schedule delays and cost overruns. This approach enables early detection when correction costs are minimal, potentially saving projects significant resources while improving overall system quality and compliance.

The solution maintains essential human oversight while dramatically reducing the manual effort required for comprehensive analysis. Systems engineers retain decision-making authority while receiving AI-generated recommendations that focus their attention on the most promising candidates, optimizing the use of scarce SE expertise.

The techniques developed are readily adaptable to other forms of requirements management challenges, including requirements validation, change impact analysis, and compliance verification. This broader applicability extends the value proposition beyond traceability analysis to comprehensive AI-augmented requirements management [7].

5. Future Work and Limitations

Current limitations include dependency on requirement quality and document structure, with performance varying based on the clarity and completeness of source materials. Future work will focus on using this solution to improve the clarity of requirements, applying the lessons learned from developing this solution to other aspects of requirement management [3].

Integration with existing SE tools and workflows represents another important development area, ensuring seamless adoption within established project environments [4]. Additionally, it is important to accept the current limitations of LLM and design solutions around them. With the next generation of LLMs, the recommendations will likely improve. It is also expected that no matter how much LLMs improve, the fundamentals of writing and managing good requirements remain essential to reduce the error of recommendation.

6. Conclusion

This research demonstrates how Large Language Models integrated with Model-Based Systems Engineering can transform traditionally labor-intensive requirements traceability analysis into an efficient, automated process while preserving essential human judgment [2]. The approach addresses real-world systems engineering challenges where manual analysis proves inadequate, offering a practical pathway for AI-augmented SE practice that scales with system complexity and project demands.

The integration of MBSE and LLM technologies represents a significant advancement in AI4SE applications, providing systems engineers with powerful tools to manage large-scale, data-intensive analysis tasks that are essential for system success but often impractical to perform manually [6]. This work contributes to the broader goal of enabling AI-enhanced systems engineering that amplifies human capability rather than replacing human expertise.

References

[1] Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M. A., Chioasca, E. V., & Batista-Navarro, R. T. (2021). "Natural language processing for requirements engineering: A systematic mapping study." ACM Computing Surveys, 54(3), 1-41.

[2] Friedenthal, S., Griego, R., & Sampson, M. (2023). "INCOSE Model-Based Systems Engineering (MBSE) Initiative." INCOSE International Symposium.

[3] Knauss, E., Liebel, G., Schneider, K., Horkoff, J., & Kasauli, R. (2020). "Quality requirements in agile as a knowledge management problem: more than just-in-time." IEEE Transactions on Software Engineering, 46(1), 96-120.

[4] Bajaj, M., Zwemer, D., Peak, R., Phung, A., Scott, A. G., & Wilson, M. (2015). "SLIM: Collaborative model-based systems engineering workspace for next-generation complex systems." IEEE Systems Conference (SysCon), 270-276.

[5] NASA. (2020). "NASA Systems Engineering Handbook." NASA/SP-2016-6105 Rev.2.

[6] Chechik, M., Nejati, S., & Sabetzadeh, M. (2019). "Software assurance in an uncertain world." In Future of Software Engineering (FOSE'19), 1-12.

[7] Henderson-Sellers, B., & Ralyté, J. (2010). "Situational method engineering: State-of-the-art review." Journal of Universal Computer Science, 16(3), 424-478.

Contact Information:

Primary Author: Henock Legesse

Email: henock.a.legesse@nasa.gov

Organization: NASA