*Proceedings of the Singapore Scientific Conference – Forging a Sustainable Future Through Science & Technology*

doi: 10.3850/081125_SSC2025_OL-04-0670

SSC2025

**SSC2025 Category/Theme:**: Manufacturing — Next-Gen Manufacturing

# AI-Assisted Decision Support Tool for Troubleshooting in Laser Manufacturing

Abdul Hady Bin Zaydie,[1] David Weidong Lin[1], and Malcolm Yoke Hean Low[2]
[1] Engineering Cluster, Singapore Institute of Technology, Singapore
[2] Infocom Technology Cluster, Singapore Institute of Technology, Singapore

Keywords: Manufacturing, Artificial Intelligence, Decision Support Systems, Failure Analysis, Operations Research

## BACKGROUND

Troubleshooting in high-mix, high-precision production is slowed by fragmented knowledge, sparse logs, and reliance on tacit experience. In our case study of a laser manufacturing workstation, we mapped 161 steps related to the assembly of the lower cavity of the laser module, of which 114 relate directly to failure generation, detection, prevention, or correction. This provides a concrete substrate to test an AI chatbot that can surface procedural knowledge, reason over multi-step dependencies, and propose auditable next steps during failure analysis (FA) and root-cause analysis (RCA).

## OBJECTIVES

We proposed and have begun preliminary development with the company's engineering team of an on-premises AI troubleshooting chatbot that: (i) answers operators query with context-aware guidance; (ii) recommends next diagnostic or corrective actions with transparent and traceable rationale; (iii) learns from outcomes to improve over time. We compare three knowledge-use paradigms – Retrieval Augmented Generation (RAG) [1], Graph-based RAG (GraphRAG) [2], and multi-Agent orchestration [3] – against usability (latency, cognitive load), retrieval/grounding quality, and actionability in real troubleshooting scenarios.

## METHODOLOGY

We implemented a web-based chatbot that runs locally on the company's hardware. Knowledge sources include work instructions, FA/RCA reports, and a step-failure taxonomy.

- RAG baseline: chunked documents with hybrid dense-sparse retrieval and prompt-level grounding [1].
- GraphRAG: a lightweight manufacturing knowledge graph (process steps, parts, symptoms, causes, controls) drive graph-constrained retrieval and path-aware reasoning to avoid hallucinated recommendations [2].
- Agents: a planner-executor setup coordinates four roles – Query Router, Retrieval, RCA Planner, and Action Verifier – drawing on agentic prompting patterns [3]. The planner integrates an Operations Research consensus layer using the Linear Ordering Problem to rank candidate actions from multiple experts and historical fixes, solved with Google OR-Tools [5].

Data-logging enhancements feed back into retrieval and post-hoc RCA. Safety rails enforce citations of sources and fallback to human escalation.

## RESULTS AND DISCUSSION

In guided sessions with the engineering team, the chatbot resolved common "what-to-check-next" queries and reduced back-and-forth with on-call engineers. RAG offered fast coverage but could miss cross-step dependencies while the GraphRAG improved disambiguation when symptoms spanned multiple stations. The multi-agent orchestration delivers stepwise, auditable guidance by aggregating expert inputs and historical fixes into a Linear Ordering consensus (via Google OR-Tools [5]) anchored to the workstation's step-failure map. Findings are consistent with reports of RAG-based troubleshooting and maintenance chatbots improving actionability in industrial contexts [4].

## CONCLUSIONS

We propose a practical framework for an AI troubleshooting chatbot that blends RAG, GraphRAG, and multi-agent planning to shift shop-floor decision support from passive document search to guided, auditable action recommendation. Early preliminary work indicates feasibility and streamlined troubleshooting process. Ongoing work scales the knowledge graph, broadens agent tool-use such as image checks, and formalizes impact via a cost model aligned to production KPIs.

## REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP," in *Advances in Neural Information Processing Systems*, 2020.
[2] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, et al., "From Local to Global: A Graph RAG Approach to Query-Focused Summarization," *arXiv:2404.16130*, 2024.
[3] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, et al., "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," *arXiv:2308.08155*, 2023.
[4] A. Narimani, "Integration of Large Language Models for Real-Time Troubleshooting in Industrial Environments based on Retrieval-Augmented Generation (RAG)," in *Proc. IEOM Int. Conf. (Germany)*, 2024.
[5] "OR-Tools Google Developers," *Google Developers*, 2019. https://developers.google.com/optimization