Applied Human Factors and Ergonomics (AHFE2025), Vol. 199, 2025, 1404–1411
https://doi.org/10.54941/ahfe1006952

MECHA: Modular Equipment Chat Helper Agent for Maintenance and Operation of Machinery Used in Heavy Equipment Production Lines

Jiale Wang, Le Ling, Wenliang Wu, and Ruiqi Lin

Dongfang Electric Digital Technology Co., Ltd., Chengdu, Sichuan 610213, China

ABSTRACT

Currently, large language models have introduced numerous new ideas for further automation in the industrial sector. The application of large language models primarily focuses on three areas: knowledge bases, workflows, and intelligent agents. For instance, the manufacturing industry has started using knowledge bases to manage the vast amount of documents generated during research and development and production processes, enabling engineers and workers to retrieve knowledge more quickly. However, due to differences in the proficiency of on-site maintenance personnel, user retrieval habits, and the limitations of information available on-site, directly constructing a knowledge base for queries cannot provide truly practical maintenance operation suggestions for on-site personnel. This study, based on the large model knowledge base and multi-agent technology, constructs an intelligent agent system for production line operation and maintenance in industrial production processes, offering applications for Q&A and multi-agent multi-turn Q&A fault diagnosis.

Keywords: Large language model, SFT, Knowledge base, Agent system, Equipment maintenance

INTRODUCTION

In industrial production, the utilization efficiency of production line equipment significantly impacts overall productivity. In the manufacturing of large-scale equipment, numerical control machines and robots used by production personnel are critical tools. However, due to challenges such as the slow training of maintenance personnel and the inconvenience of consulting on-site operation and maintenance manuals, production losses caused by downtime have remained persistently high. According to statistics, machine tools experience approximately 1,500 hours of unplanned downtime annually. If a factory has an annual output value of 100 million yuan, this translates to an estimated loss of 10.27 to 17.12 million yuan. Improving equipment maintenance efficiency can enhance OEE (Overall Equipment Effectiveness), thereby generating more revenue for the factory.

Implementing a production line operation and maintenance system can save approximately 40% of maintenance personnel's troubleshooting time,

50% of new employee training time, and 90% of work order documentation time. Assuming on-site maintenance personnel spend 70% of their time on troubleshooting, 20% on submitting work orders, and 10% on training, comprehensive calculations suggest that such a system could reduce unplanned downtime by 30% to 50%.

Current operation and maintenance systems often remain at a digitalization stage that only provides data monitoring, lacking output and guidance capabilities, thus representing a non-intelligent phase. Leveraging large language models, operation and maintenance systems can acquire the ability to "converse" with production line maintenance workers and contribute to OEE improvement through various means, such as providing data analysis support, enabling rapid queries of equipment fault manuals, facilitating early-stage maintenance training, automating work order completion and transmission, and guiding equipment inspections. Specifically, we have developed a multi-agent operation and maintenance system to realize these functions.

Contributions of this paper:

- Provides a holistic and validated system architecture solution and implementation pathway for LLM-assisted production lines.
- Proposes a multi-tier, multi-agent business architecture design for multi-turn dialogues in maintenance scenarios.
- Completes the fine-tuning of a compact 7B model (including parts of publicly available datasets) for this specific scenario.

DMG MORI SERIES DATA

For this scenario, DMG series machine tools were selected for the experiment, specifically the DMG MORI DMU 50 five-axis machining center. The primary materials used are the official supporting DMG product manual and maintenance manual. The product manual helps the large language model understand the machine tool's structure and, in this project, serves as the Q&A data for fine-tuning the model. The maintenance manual, which provides reference for troubleshooting methods across different processes, is used as the vectorized content for the RAG-based knowledge base.

Table 1: Data source.

Purpose	Source	Number of Chunks	Factual Data Items
Fine-tuning	Common Fault Diagnosis and Maintenance of Precision Machine Tools Common Fault Diagnosis and Maintenance of Milling Machines Common Fault Diagnosis and Maintenance of Machine Tool Hydraulic Systems Common Fault Diagnosis and Maintenance of Lathes	3066	11406

Continued

Table 1: Continued

Purpose	Source	Number of Chunks	Factual Data Items
	Common Fault Diagnosis and Maintenance of Grinding Machines DMG MORI DMU50 5-Axis Machining Center User Manual (Chinese Version)		
RAG	Sinumerik 840D_840Di_810D Diagnostic Manual	932	\

Fine-Tuning

In this project, the machine tool's product manual was utilized as the core training data for fine-tuning the large language model. This process is crucial as it enables the model to develop a deep understanding of the complex structure, working principles, and interactions between various components of the DMG MORI DMU 50 five-axis machining center. By fine-tuning the model parameters, we ensured that the model can provide accurate and specific information aligned with this particular machine tool model when addressing questions regarding its characteristics, operational protocols, and potential failure modes. Consequently, all our intelligent agents are equipped with fundamental knowledge about this specific machine tool.

Regarding data processing, we employed a two-stage approach to generate question-answer pairs for fine-tuning the large model. Since the user manual consists of unstructured data, the first step involved using an LLM to extract objective information from it. This extracted information underwent manual verification to ensure factual consistency with the original text, regardless of subsequent data formatting. The processed data was then organized into individual, discrete declarative sentences. In the second step, an LLM was used to generate question-answer pairs from these sentences. Typically, 3 to 1 QA pairs were generated from several sentences, forming the final dataset used for fine-tuning. Using this methodology, we obtained a total of 1,409 question-answer pairs.

Furthermore, to enhance the model's transferability to other types of automated machining equipment, we applied the same data extraction process to a series of books authored by the Editorial Committee of Machine Tool Fault Diagnosis and Maintenance: Common Fault Diagnosis and Maintenance of Precision Machine Tools, Common Fault Diagnosis and Maintenance of Milling Machines, Common Fault Diagnosis and Maintenance of Machine Tool Hydraulic Systems, Common Fault Diagnosis and Maintenance of Lathes, and Common Fault Diagnosis and Maintenance of Grinding Machines. These five books provided us with an additional 8,395 question-answer pairs.

Parts of the relevant datasets have been publicly released on Hugging Face. Based on the aforementioned data, we performed a full parameter fine-tuning of a 7B model using two A100 GPUs.

Before and after fine-tuning, 200 data entries were randomly selected for validation, and the evaluation results of the large model were manually scored. The results are as follows:

- Evaluation of the large model without fine-tuning: The score for the 200 entries was 7.7.
- Evaluation of the fine-tuned large model: After fine-tuning, the loss decreased from 2.0 to 0.2 over 700 steps. The score for the 200 entries was 8.5.

Knowledge Base

When building the RAG (Retrieval-Augmented Generation) knowledge base, we utilized the maintenance manual provided by DMG MORI. This manual primarily contains maintenance information corresponding to each error code. First, the manual content is transformed into structured data, including information such as error codes, fault phenomenon descriptions, probable causes, recommended solutions, and required tools. Subsequently, an embedding model is used to vectorize this information, enabling the large language model to efficiently retrieve and integrate relevant content. During the vectorization process, this information is grouped and chunked, with all content corresponding to a single error code treated as one group. This ensures the model retrieves complete information about a fault from the database.

This test compared the recall rate of knowledge retrieval in the industry with and without a knowledge base. The comparison results are as follows:

- For 200 questions that had identical matches in the knowledge base, the answers were largely correct.
- For 43 questions with similar but not identical matches in the knowledge base, the responses leveraged relevant answers from analogous cases. When the solutions for similar issues were generally consistent, the answers remained reasonably accurate; however, when the solutions for similar problems differed significantly, the responses showed noticeable deviations.
- For 43 questions with no similar matches in the knowledge base, the answers relied entirely on the large language model, resulting in more generalized responses.

This test compares the recall rates under different top-k values, illustrating the changes in recall rate and average top-k performance as the k-value increases.

MULTI-LAYERED MULTI-AGENT FRAMEWORK

We have designed a specialized multi-turn dialogue system for this scenario. Based on our research into the actual maintenance operations performed by on-site workers, the troubleshooting process typically begins with using error codes and monitoring data to make an initial assessment of the faulty component. Subsequently, the corresponding manual is consulted to locate the inspection procedures for that specific component. During the inspection,

maintenance personnel must refer to the Operation Manual and Maintenance Manual based on the current situation, making real-time judgments to determine the fault type, select the appropriate tools, and complete the repair.

It is evident that identifying the faulty component and performing the inspection are two distinct stages. Furthermore, based on practical feedback, if the initial inspection reveals no fault in the suspected component, the process must return to the previous step of re-identifying the potential faulty part. Additionally, during the inspection phase, the methods and operational procedures differ depending on the component being checked. The multi-agent system must, therefore, prompt the maintenance personnel for different feedback based on the specific component under investigation.

Our multi-agent system is designed to directly mirror this maintenance process: it first identifies the component requiring maintenance, then determines the fault type, and finally collaborates with the maintenance personnel to complete the repair. To achieve this, we implemented a multi-tiered multi-agent system. Unlike traditional multi-agent systems that feature only a single master Agent, our system incorporates an additional managing Agent at each tier, responsible for overseeing tasks at that specific level. The final architecture is illustrated in the figure below. The Querying Agent and the Knowledge Retrieval Agent are common across different layers, and all Agents share a unified memory stack. This Multi-Agent system was implemented using the open-source framework, Shannon, for application development.

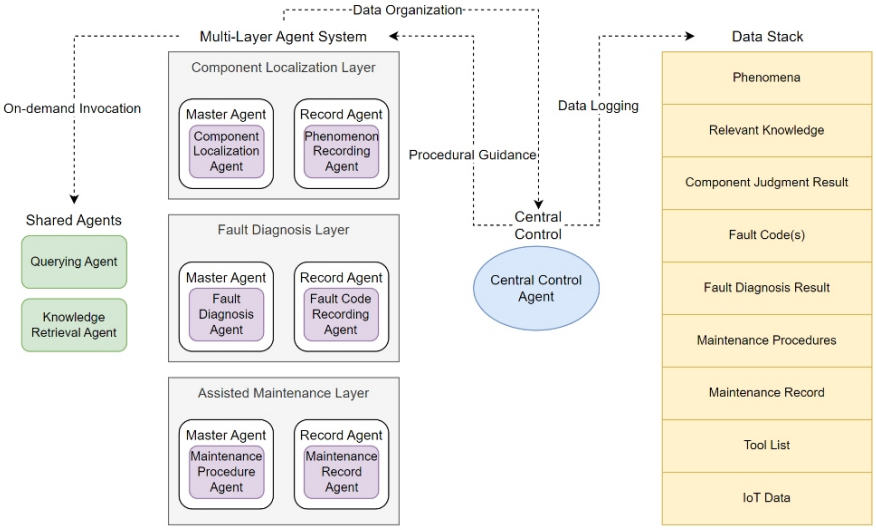

Figure 1: System architecture.

Component Localization Layer

In the Component Localization Layer, the master Agent is the Component Localization Agent, assisted by the Phenomenon Recording Agent, Querying Agent, and Knowledge Retrieval Agent. At this level, the Phenomenon

Recording Agent extracts and refines the descriptive information provided by the user, eliminating ambiguous expressions while converting image and audio data into text descriptions. These processed descriptions are then transmitted to the master Agent, which subsequently invokes the Knowledge Retrieval Agent. Based on the master Agent's preliminary judgment and the recorded phenomena, the Knowledge Retrieval Agent retrieves relevant information from the fault diagnosis knowledge base. If the master Agent determines that the current information is insufficient to identify the faulty component, it will notify the user and invoke the Querying Agent to obtain additional relevant information. When the master agent assigns a high confidence score to its judgment, it preliminarily identifies the faulty component and proceeds to the next processing stage.

Fault Diagnosis Layer

In the Fault Diagnosis Layer, the master Agent is the Fault Diagnosis Agent, assisted by the Fault Code Recording Agent, Querying Agent (shared across layers), and Knowledge Retrieval Agent (shared across layers). The Fault Code Recording Agent extracts fault codes from relevant recorded phenomena. If no fault code is detected, it notifies the master Agent, which then uses the Querying Agent to obtain fault code information from the user. Upon acquiring the fault code, the system combines it with the recorded phenomena and utilizes the Knowledge Retrieval Agent to query relevant records from the manual in the knowledge base. If no fault code is confirmed, the retrieval is performed directly based on the phenomenon descriptions. This layer provides the user with preliminary diagnostic results along with confidence scores, while storing the retrieved relevant maintenance information.

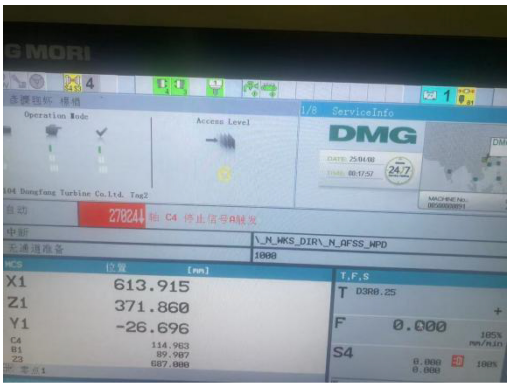

Figure 2: The fault code.

Component Localization Layer

In the Assisted Maintenance Layer, the master Agent is the Maintenance Procedure Agent, supported by the Maintenance Record Agent, Querying Agent, and Knowledge Retrieval Agent. The master agent extracts the required maintenance procedures and tool list from the information received in the previous step and presents them to the user. The user can then report

their maintenance progress to the master Agent. At any point, if the master Agent determines that it cannot ascertain the user's maintenance progress, it will instruct the Querying Agent to pose questions to the user. If during the maintenance process the user indicates that the faulty component or root cause was incorrectly diagnosed, the central control Agent will command the system to return to the respective layer for re-diagnosis. Ultimately, if the maintenance is completed, all information from the data stack, along with IoT data, will be stored as a case.

SYSTEM INTEGRATION

The multi-agent maintenance system is deployed on servers located within the factory premises. On-site workers can connect to these servers and access data using 5G tablets. To simplify system usage for field personnel, we integrated this multi-agent framework with on-site IoT devices, cameras, and audio equipment. By incorporating TTS (Text-to-Speech) and VLM (Vision-Language Models) capabilities, the system enables users to capture real-time photos and use voice messages for interaction. Furthermore, this integration allows the system to upload IoT data alongside diagnostic results, facilitating data collection. This collected data can subsequently be used to train small models for IoT threshold judgment. Ultimately, all these modules are integrated into the application.

CONCLUSION

This paper proposed MECHA, a knowledge understanding and intelligent Q&A system for the operation and maintenance of large-scale production line equipment. The system is designed to address core challenges in industrial settings, such as low maintenance efficiency and prolonged unplanned downtime, which stem from variations in personnel skills, difficulties in information retrieval, and complex fault diagnosis procedures.

Through preliminary validation in a real industrial scenario, the MECHA system has demonstrated its significant potential for practical application in saving maintenance time, improving diagnostic efficiency, and reducing unplanned downtime. This research provides an effective and implementable paradigm for the application of LLM and multi-agent technologies in complex industrial environments.

Looking ahead, we will further explore the system's transferability to a wider range of equipment types, optimize the understanding and utilization of multi-modal information, and investigate how to leverage the vast amount of case data generated during system operation to enable autonomous and dynamic optimization of fault prediction and maintenance strategies, ultimately pushing industrial maintenance towards a higher level of intelligence.

REFERENCES

Kocoro-lab. (n.d.). Shannon: Kubernetes/Linux of AI Agents - An enterprise-ready AI agent platform built with Rust for performance, Go for orchestration, Python for LLMs, and Solana for Web3 trust. GitHub. Retrieved from https://github.com/Kocoro-lab/Shannon.

Li, G., Hammoud, H., Itani, H., Khizbullin, D., & Ghanem, B. (2023). CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. In Proceedings of the Neural Information Processing Systems.

Sanwal, M. (2025). Layered Chain-of-Thought Prompting for Multi-Agent LLM Systems: A Comprehensive Approach to Explainable Large Language Models. arXiv preprint arXiv:2501.18645.

Wang, L., Ma, C., Feng, X., et al. (2024). A survey on large language model based autonomous agents. Frontiers of Computer Science, 18, 186345.

Xu, W., Deutsch, D., Finkelstein, M., Juraska, J., Zhang, B., Liu, Z., Wang, W. Y., Li, L., & Freitag, M. (2024). LLMRefine: Pinpointing and Refining Large Language Models via Fine-Grained Actionable Feedback. In Findings of the Association for Computational Linguistics: NAACL 2024 (pp. 1429–1445). Mexico City, Mexico: Association for Computational Linguistics.

Zhang, M., Zhou, J., Meng, F., Zeng, J., Liu, X., Wang, Y., & Wong, D. F. (2025). DELTA: An Online Document-Level Translation Agent Based on Multi-Level Memory. arXiv preprint arXiv:2410.08143.