Intelligent Fault Diagnosis of Cyber Physical Systems using Knowledge Graphs

Ameneh Naghdi Pour[1], Benno Kruit[1], Jieying Chen[1], Peter Kruizinga[2], Godfried Webers[3] and Stefan Schlobach[1]

[1]Vrije Universiteit Amsterdam, De Boelelaan 1111, 1081 HV Amsterdam, NL
[2]Canon Production Printing Netherlands, Van der Grintenstraat 1, 5914 HH Venlo, NL
[3]Philips Medical Systems International, Veenpluis 6, 5684 PC Best, NL

Abstract

Machine maintenance poses significant challenges and costs in equipment manufacturing. A considerable portion of the budget is dedicated to provide training materials (documentations) for service engineers to diagnose failure causes, as well as to cover their salaries and provide spare parts. Additionally, breakdowns adversely impact machine capacity, preventing customers from utilizing their equipment during downtime. The main reason for all these challenges is the lack of efficiently utilizing training documentations. To address this and reduce costs while mitigating the negative implications of machine breakdowns, we propose a two-phase framework consisting of knowledge graph construction and diagnosis. In the first phase, an upper-level ontology based on the requirements for fault diagnosis of Cyber Physical Systems is developed, by drawing inspiration from the Industrial Domain Ontology (IDO) and the Industrial Ontology Foundry-Maintenance Reference Ontology (IOF-MRO). In the second phase, SPARQL queries are executed on the knowledge stored in GraphDB, providing valuable insights for diagnosing machine failures.

Keywords

Fault diagnosis, knowledge graphs, ontology engineering

1. Introduction

Maintenance tasks in equipment manufacturing encounter several challenges. There is a considerable financial burden on both manufacturers and customers, stemming from expenses related to training service engineers, their salaries, and spare parts. Additionally, downtime negatively impacts machine capacity, preventing customers from utilizing their equipment during these intervals. Canon and Philips, our industrial partners for this project, assist with maintenance tasks by training their service engineers and providing them with documentation and occasional video resources. However, navigating extensive documentation can be challenging, and the costs associated with producing training videos present additional difficulties.

To address these challenges and enhance support for service engineers in fault diagnosis of Cyber-Physical Systems (CPS), we are developing a framework based on qualitative knowledge-driven fault diagnosis. It is important to highlight that our industrial partners are engaged in the production of large-scale CPS, such as printers and MRI machines that comprise thousands of parts, sensors, actuators, and computer interfaces, leading to a high level of complexity.

Figure 1 shows our proposed framework in which there are two main phases along with the input sources and output results. The input consists of three categories of data and knowledge identified through interviews conducted with authorities in the manufacturing field. In the first phase, we will

EKAW 2024: EKAW 2024 Workshops, Tutorials, Posters and Demos, 24th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2024), November 26-28, 2024, Amsterdam, The Netherlands.
a.naghdipour@vu.nl (A. Naghdi Pour); b.b.kruit@vu.nl (B. Kruit); j.y.chen@vu.nl (J. Chen); peter.kruizinga@cpp.canon (P. Kruizinga); godfried.webers@philips.com (G. Webers); k.s.schlobach@vu.nl (S. Schlobach)
https://https://github.com/Ameneh71 (A. Naghdi Pour); https://github.com/bennokr/ (B. Kruit); https://github.com/JieyingChenChen (J. Chen)
0000-0002-7357-7906 (A. Naghdi Pour); 0000-0002-3282-1597 (B. Kruit); 0000-0002-3282-1597 (J. Chen); 0000-0002-3282-1597 (P. Kruizinga); 0000-0002-3282-1597 (S. Schlobach)

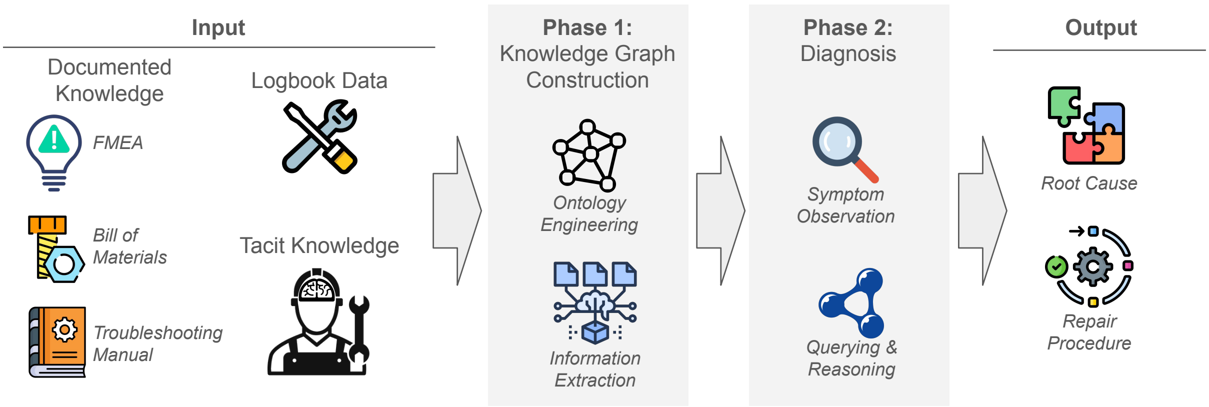

Figure 1: Construction and application framework of the domain fault knowledge graph.

construct a Knowledge Graph (KG) based on the gathered input. This involves manually creating an upper-level ontology, inspired by IDO [1] and IOF-MRO [2], which will serve as the foundation for the KG. In the subsequent phase, service engineers will identify a symptom that needs to be translated into a query for the KG reasoning. The KG will then support a root cause analysis and provide the corresponding repair procedure, which will be presented to the service engineer as the output.

2. Related work

Various knowledge sources support service engineers in diagnosing failures, including machine log data, documented knowledge, logbooks from engineers, and expert tacit knowledge. Researchers have explored these sources and proposed methods for maintenance tasks, grouped into four categories: (1) Model-based approach: Constructs a physical model to compare actual system output with predicted values, using consistency monitoring. This method is accurate but requires precise physical models [3]. (2) Signal-based approach: Analyzes monitored signals to diagnose faults but is limited to key sensor-equipped components [4]. (3) Quantitative knowledge-based method: Treats diagnosis as a pattern recognition problem using historical data but requires substantial fault data [5]. (4) Qualitative knowledge-based method: Uses a qualitative model and techniques like searching, matching, and reasoning without needing physical models or extensive data, relying instead on a fault knowledge base [6].

Given the need for system-level analysis and reasoning, the qualitative knowledge-based method is ideal for this project. A key aspect of this method is the need for a model to represent fault knowledge, with commonly utilized models including Rule-Based [7], Fault Trees [8], and Petri Nets [9]. However, these models have limitations, such as requiring prior fault analysis and manual editing, making updates difficult [6]. This paper proposes using knowledge graph technology to mine fault knowledge and create a structured fault knowledge base.

3. Methodology

Figure 1 presents our proposed framework for fault diagnosis of Canon and Philips CPS. Below, we will outline the framework in three main subsections.

3.1. Input Source

Following regular interviews with our partners, we have identified various maintenance sources and categorized them into three groups: Data, Documented Knowledge, and Tacit Knowledge. Each of these sources offers unique insights and information about the system, but they also have limitations.

- Data: This includes logbooks that provide insights into problems and the actions taken to resolve them. However, this information can be incomplete, as service engineers might only record "done" without detailing the action taken.
- Documented Knowledge: This category encompasses three resources including (1) Bill Of Material, which provides information about the physical structure of the system, but lacks details on problems that may happen for each part of the system. (2) Failure Mode Effect Analysis, provides insights into challenging failure modes and their potential effects, but it focuses solely on complex issues. (3) Troubleshooting Manuals, offers information on the causes of failures and remedies for them.
- Tacit Knowledge: This refers to undocumented knowledge that resides in the minds of experts, making it challenging to acquire.

These sources are complementary; by integrating them, we can leverage their strengths and compensate their limitations.

3.2. Knowledge graph construction

The first phase involves knowledge graph construction, which encompasses two primary steps: upper-level ontology construction and information extraction. The following subsections provide a detailed explanation of these two steps.

3.2.1. Upper-level Ontology Construction

We manually developed an upper-level ontology as the foundational structure for organizing the schema of the knowledge graph. This process involved thorough analysis of various input sources to pinpoint the most valuable knowledge. We also formulated competency questions to highlight key queries for the knowledge graph and conducted interviews with industrial partners to align their expectations with the ontology. Our fault diagnosis ontology was also compared with the IDO and IOF-MRO. The design of the ontology is depicted in Figure 2 , which comprises classes and relationships representing the domain of interest. There is a class named "`Component`" that can represent a part, assembly, or subsystem, each serving a specific "`Function`". When a part encounter malfunction, it leads to the occurrence of a "`Problem`". The problem has cause(s), which can be considered as problem, result in a certain "`Effect`". To address these problems, there are two types of "`Procedure`" that can be implemented. The first is a "`Solution`", which directly solves the problem at hand. The second is a "`Workaround`", which aims to address the effects caused by the problem. Both procedures consist of multiple "`Step`", with each step involving a specific part of the system. Due to space constraints, we are unable to provide additional information on the definitions of these entities and their comparison with the IDO and IOF-MRO ontologies.

3.2.2. Information Extraction

In this step, we employed various techniques including Regular Expressions, Named Entity Recognition, and Large Language Models (LLMs), to extract the necessary entities from the identified sources. These techniques allowed us to systematically identify relevant information, which we then organized into structured tables. A key challenge in our current data sources was the lack of information regarding functions, dependencies, and causes associated with the entities. To bridge this gap, we leveraged the capabilities of LLMs by treating them as expert engineers. We provided the model with information about the system's structure and posed specific questions related to its functions, dependencies, and causes. This approach enabled us to gather insights that were previously unavailable. Once we acquired the necessary information, we transformed the structured tables into RDF (Resource Description Framework) triples. This conversion facilitated the integration of our data into a graph database, allowing for enhanced querying and analysis of the relationships between various entities within the system.

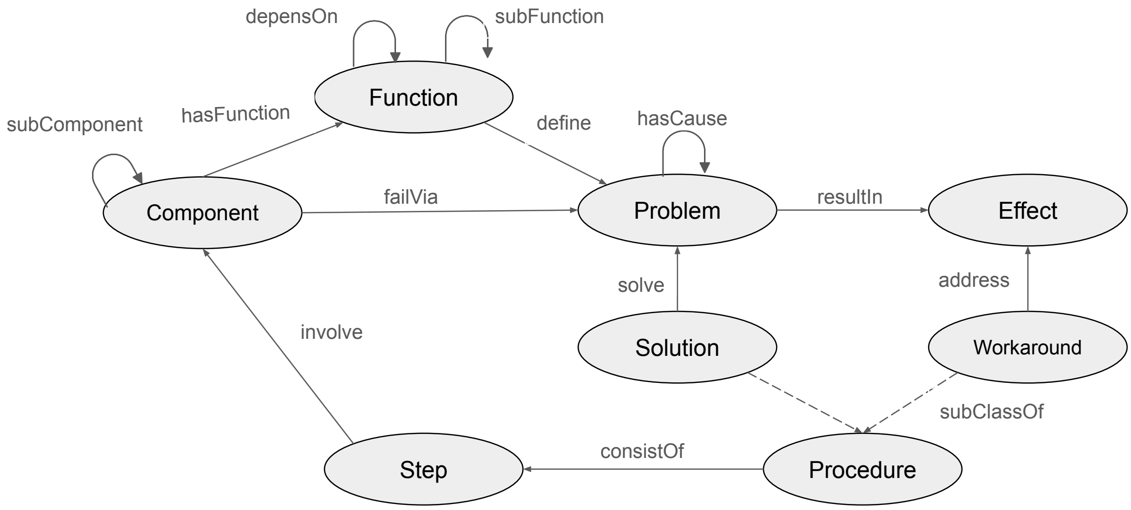

Figure 2: Zorro upper-level ontology; modeling physical and functional views as well as causal and dependency relations.

3.3. Diagnosis

The next phase, diagnosis, illustrates the application of our proposed method, wherein service engineers observe symptoms of a failure that need to be transformed into queries for knowledge graph-based reasoning. This process will yield a diagnosis identifying the root cause of the issue, along with a suggested procedure for repair. To achieve this, the knowledge extracted in the previous phase must be organized and stored in a specific structure. In this paper, we utilize graphDB [10], one of the leading graph database software, which is designed to store data in a graph structure based on the RDF model. Once the data is stored in graphDB, we execute SPARQL queries over the knowledge graph. SPARQL, a powerful query language for RDF data, enables us to retrieve and manipulate the data effectively. In this setup, RDFS (RDF Schema) serves as the reasoning layer, providing semantic inference capabilities, while SPARQL acts as the querying interface for accessing the knowledge graph.

4. Conclusion and Future work

In this study, we developed a framework for fault diagnosis based on Knowledge Graph. First, we manually created an upper-level ontology to serve as the schema for the knowledge graph. We then employed various techniques to extract knowledge and populate the graph according to this ontology. Subsequently, we implemented a straightforward approach utilizing RDFS reasoning and SPARQL queries to perform diagnosis. This allowed us to analyze the data within the knowledge graph effectively. For the future work, we will focus on enhancing the reasoning capabilities of the framework by integrating additional techniques, such as Bayesian Networks. This refinement aims to improve the accuracy and depth of the diagnostic process.

Acknowledgments

This publication is part of the project ZORRO with project number KICH1.ST02.21.003 of the research programme Key Enabling Technologies (KIC) which is (partly) financed by the Dutch Research Council (NWO).

References

[1] ISO, ISO 15926-14:2020(E), ISO 15926 Part 14: Industrial top-level ontology, Technical Report, ISO, Geneva, CH, 2020.

[2] M. Hodkiewicz, C. Woods, M. Selway, M. Stumptner, Industrial ontology foundry (iof) maintenance reference ontology, 2024. doi:10.26182/chzp-vs60.

[3] S.-H. You, Y. M. Cho, J.-O. Hahn, Model-based fault detection and isolation in automotive yaw moment control system, International Journal of Automotive Technology 18 (2017) 405–416. doi:https://doi.org/10.1007/s12239-017-0041-5.

[4] K. Zhang, Y. Xu, Z. Liao, L. Song, P. Chen, A novel fast entrogram and its applications in rolling bearing fault diagnosis, Mechanical Systems and Signal Processing 154 (2021) 107582. doi:https://doi.org/10.1016/j.ymssp.2020.107582.

[5] J. Dong, D. Su, Y. Gao, X. Wu, H. Jiang, T. Chen, Fine-grained transfer learning based on deep feature decomposition for rotating equipment fault diagnosis, Measurement Science and Technology 34 (2023) 065902. doi:https://doi.org/10.1088/1361-6501/acc04a.

[6] X. Tang, G. Chi, L. Cui, A. W. Ip, K. L. Yung, X. Xie, Exploring research on the construction and application of knowledge graphs for aircraft fault diagnosis, Sensors 23 (2023) 5295. doi:https://doi.org/10.3390/s23115295.

[7] Z. Wang, S. Li, W. He, R. Yang, Z. Feng, G. Sun, A new topology-switching strategy for fault diagnosis of multi-agent systems based on belief rule base, Entropy 24 (2022) 1591. doi:https://doi.org/10.3390/e24111591.

[8] K. Pan, H. Liu, X. Gou, R. Huang, D. Ye, H. Wang, A. Glowacz, J. Kong, Towards a systematic description of fault tree analysis studies using informetric mapping, Sustainability 14 (2022) 11430. doi:https://doi.org/10.3390/su141811430.

[9] C. Xu, J. Li, X. Cheng, Comprehensive learning particle swarm optimized fuzzy petri net for motor-bearing fault diagnosis, Machines 10 (2022). doi:ttps://doi.org/10.3390/machines10111022.

[10] Ontotext, Graphdb - rdf triplestore semantic graph database, https://www.ontotext.com/products/graphdb/, 2023. Accessed: 2024-09-13.