2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
July 15-19, 2024. Boston, MA, USA

Description Method and Failure Ontology for Utilizing Maintenance Logs with FMEA in Failure Cause Inference of Manufacturing Systems

Takuma Fujiu[1], Toshinori Yasui[2], Sho Okazaki[1], Kohei Kaminishi[3], and Jun Ota[3]

Abstract— **In the realm of manufacturing systems, inferences of failure causes have been performed mainly depending on the reuse of Failure Mode and Effect Analysis (FMEA). However, achieving inference with the same level of quality as experts has been challenging. The objective of this study is to improve the quality of inference by combining maintenance logs and FMEA in the inference of failure causes of manufacturing systems. There are two challenges in using maintenance logs for inference. First, it is difficult to extract causal relationships among failures because the format and quantity of maintenance logs are not consistent. Second, the hierarchy of failures described in each item of maintenance logs is not fixed, making it difficult to search for similar failures by using the items. To address these challenges, we propose a description method for maintenance logs that describes causal relationships among failures and those between failures and functions. We also introduce a failure ontology that represents the hierarchy of failures and conditions impaired by failures based on expert knowledge of manufacturing systems. For the two assumed failures, the inferences derived from maintenance logs and FMEA using the proposed method show better quality than the inferences using FMEA alone. The recall calculated from the failure cause candidates enumerated by experts increased by 3.7 and 5.0 times, respectively.**

I. INTRODUCTION

To maintain and improve the productivity of manufacturing systems, investigating the causes of failures is crucial. In the investigation, after enumerating possible causes for the occurred failure, the true cause is identified by checking each of them one by one. For example, when assembly misalignment occurs in the assembly process, potential causes such as conveyor wear and gripper deterioration are enumerated and checked. Although enumerating possible failure causes is an important task, it is difficult for non-experts without knowledge and experience to conduct this task. The Japanese manufacturing industry is facing a shortage of experts, and methods to assist non-experts in investigating failure causes are receiving more attention. For the enumeration process, a practical approach is to refer to analyses that have been performed by experts in the past. FMEA and maintenance logs are typical examples of failure analysis data that are available in factories. Failure Mode and Effect Analysis (FMEA) is an analysis of the causes, effects, and existing control methods for each of the potential failure modes listed as far as possible[1]. FMEA involves a hierarchical analysis of the structure of the target system to enumerate potential failures and is conducted by a team of experts. Maintenance logs are data analyzed for actual failures and contain more detailed descriptions than FMEA. Since there are some manufacturing systems with similar processes, it is important to use FMEA and maintenance logs on similar systems to infer failure causes.

There have been some attempts to reuse FMEAs. A typical method to retrieve information in FMEA is to use ontology. Ontology is a formal description of all the entities of a domain and the relations existing between these entities[2]. Ontologies on FMEA were constructed and past cases were retrieved using queries[3], [4], [5]. Another method of retrieving FMEAs is case-based reasoning (CBR). CBR is a method of reasoning based on similarities with past cases in solving new problems and enables the solution of new problems by repeating the cycle of retrieval, reuse, revision, and retention[6]. Mikos et al. defined some features in the FMEA content, determined the similarities to the input, and searched for similar past cases[7]. Okazaki et al. proposed a framework of failure cause inference using past FMEAs on other manufacturing systems by combining ontologies and CBR[8]. This involves constructing an ontology from the structure of the FMEA, calculating the similarity between past cases and input failures based on this ontology and the model of the target system, and enumerating potential failure causes. However, experts pay more attention to the details of the manufacturing system than FMEA during improvement activities[8], suggesting the need to use maintenance logs in addition to FMEA for failure cause inference in manufacturing systems.

Devaney et al., who conducted a search of maintenance logs using case-based reasoning (CBR), cited difficulties in the reuse of maintenance logs, such as unformatted input and the use of technical terms[9]. Due to the difficulties of reusing maintenance logs, failure cause inference using maintenance logs of similar systems has not been performed.

The objective of this study is to improve the quality of failure cause inference in manufacturing systems by using maintenance logs in conjunction with FMEAs. There are two challenges to overcome. First, the current maintenance logs do not have a unified description format for reuse. This causes variation in the amount and content of the descriptions, which leads to a lack of information necessary for inference, and also makes it difficult to extract causal

*This work was supported by New Energy and Industrial Technology Development Organization (NEDO) under the grant program JPNP18002

[1]Takuma Fujiu and Sho Okazaki are with Department of Precision Engineering, School of Engineering, The University of Tokyo, 113-8656, Tokyo, Japan

[2]Toshinori Yasui is with DENSO CORPORATION, 448-8661, Aichi, Japan

[3]Kohei Kaminishi and Jun Ota are with Research into Artifacts, Center for Engineering, School of Engineering, The University of Tokyo, 113-8656, Tokyo, Japan

relationships among the descriptions. Second, unlike FMEA, the hierarchies of failures described in maintenance logs are not organized by item. Because FMEAs are conducted by hierarchically analyzing the structure of the target systems, the hierarchies of failures described for each item are clear. On the other hand, maintenance logs do not specify which hierarchy to focus on when describing failures, making it difficult to extract similar failures by using the items. To solve these problems, we propose two approaches. The first approach is to propose a description method of maintenance logs that describes the cause-and-effect relationship of failures and the relationship between failures and functions. The second approach is to construct a failure ontology that expresses the hierarchies and characteristics of failures based on expert knowledge.

II. METHOD

A. Overview of Proposed Framework

Fig. 1 shows the overview of the proposed framework. This is an improvement of the existing framework for failure cause inference[8] to use maintenance logs in conjunction with FMEA. As shown in yellow in Fig. 1, the maintenance logs recorded in the description method proposed as the first approach are stored in the maintenance log database and converted into the past cases ontology with FMEA by the structuring module. The second approach, the failure ontology, is constructed as part of the domain ontology, as shown in green in Fig. 1.

The three databases used for inference are the FMEA database, the maintenance log database, and the SysML database, which are provided through the administrator UI. The domain ontology defines and structures knowledge and concepts about manufacturing systems, and it explains the contents of the FMEAs and maintenance logs. In the domain ontology, the classes and properties defined by Okazaki et al[8] are used in this framework. The SysML database contains diagrams that represent actions and states of the target manufacturing system, described by SysML. SysML (System Modeling Language) is a graphical modeling language that supports the analysis of complex systems, such as manufacturing systems[10]. The SysML database is converted into the process order model which represents the sequential relationships between actions and processes of the system by the process ordering module. Through the UI for users, users input the failure that occurred and the process where it occurred and receive a list of candidate failure causes as an inference output. Failure cause inferences are performed using the past cases ontology, the domain ontology, and the process order model.

B. Description Method of Maintenance Logs

1) Format of Maintenance Logs: Since the objective is to use maintenance logs in conjunction with FMEAs, the description format for maintenance logs is proposed as an extension of the FMEA format. Table I shows the proposed format of maintenance logs and an example of the description. The proposed format of maintenance logs describes the cause-and-effect relationship of failures sequentially, with three items: "Failure Phenomenon," "Failure Cause," and "Causal Factor". This allows us to write down the propagation of failures regardless of the hierarchies where failures occur and to extract causal relationships in inferring the failure causes. In addition, causal relationships between "Failure Phenomenon" and "Failure Cause" and between "Failure Cause" and "Causal Factor" can be stored, respectively, and this enables us to handle inputs at multiple levels in inferring the failure causes. To enhance the reusability of maintenance logs, we also propose to describe the functions that should have been achieved at the point where the failures occurred. This makes it possible to reuse maintenance logs not only for the same system but also for similar systems through the comparison of functions.

2) Construction of Past Cases Ontology: In order to use the FMEA database and the maintenance log database for failure cause inference, the past cases ontology structures the relationship between the descriptions. As the classes of the past cases ontology, "FailureElement" which represents failures, and "Function" which represents functions are defined. As the subclasses of "FailureElement", "FailurePhenomenon", "FailureCause", and "CausalFactor" are defined, corresponding to Table I. As the properties,

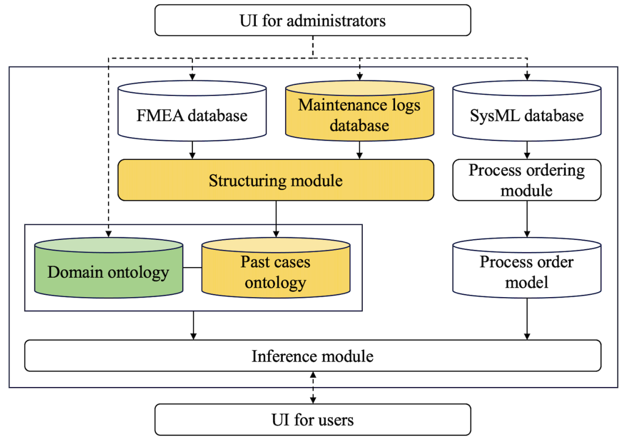

Fig. 1. Overview of the proposed framework

TABLE I
FORMAT OF MAINTENANCE LOGS

	Example	**Correspondence with FMEA**
Function	assemble a chip	
Failure Phenomenon	assembly misalignment	**Effect**
Function	chuck a chip	**Function**
Failure Casue	chuck misalignment	**Failure Mode**
Function	move to chuck position	
Causal Factor	robot's positional displacement	**Cause**

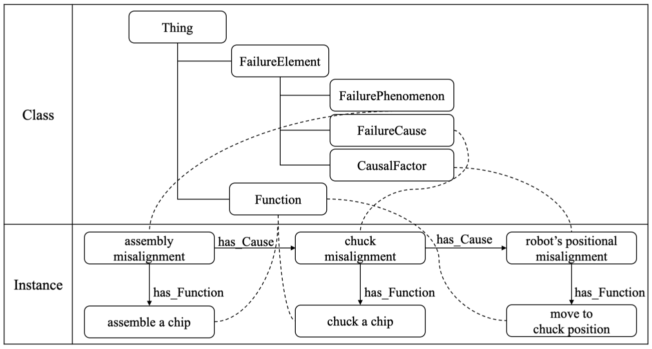

Fig. 2. The descriptions in FMEA and maintenance logs are represented as instances of the past cases ontology.

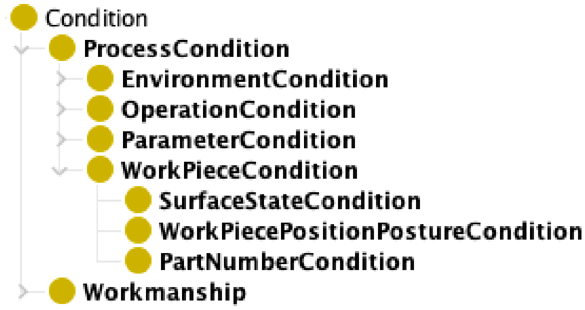

Fig. 3. "Condition" has "Workmanship" and "ProcessCondition" as its subclasses. Within each of them, conditions are structured based on their respective natures.

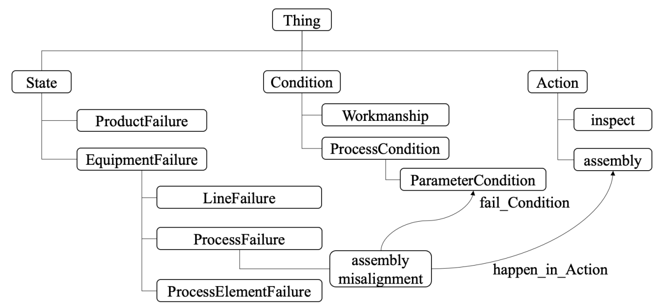

Fig. 4. An example of "Assembly Misalignment" represented by the failure ontology.

"has_Cause" which represents the causal relationship between failures, and "has_Function" which represents the relationship between failures and functions are defined.

The descriptions stored in the FMEA database and the maintenance log database are defined as instances of the past cases ontology, as shown in Fig. 2. FMEA is represented in the same ontology with maintenance logs by the correspondences between items shown in Table I.

C. Failure Ontology

In manufacturing systems, where products undergo various processes on the production line, failures can occur not only in the products but also in the equipment. This complexity poses a challenge when using existing ontologies, which typically focus on representing product failures alone. Therefore, constructing a failure ontology based on experts' knowledge is essential to represent failures in manufacturing systems. The knowledge is extracted from conversations with experts and documents within an automobile parts manufacturer.

1) Experts' knowledge of Failure: Failures in manufacturing systems are broadly classified into product failures and equipment failures. Furthermore, equipment failures are classified hierarchically into line failures, process failures, and process element failures, representing high-level, middle-level, and low-level, respectively. When a failure occurs in one level of the hierarchy, the cause exists in the same level or a lower level.

Products and processes have "workmanship" and "process conditions" that must be satisfied under normal states, respectively. States in which these conditions are not satisfied are recognized as product failures and process failures, respectively. Process element failures are broadly classified into physical factors and human factors. Within each factor, there are hierarchical relationships based on the level of abstraction of expression. For example, the sub-concepts of "corrosion" include "oxidation" and "electrical corrosion. Line failures include cycle time delays and equipment stoppages.

2) Construction of Failure Ontology: The failure ontology is built within the domain ontology that explains the concepts of manufacturing systems in the past cases ontology. In the domain ontology, the concepts of failure exist in the "State" class. As mentioned above, under the "State" class, "ProductFailure" and "EquipmentFailure" are defined. "LineFailure", "ProcessFailure", and "ProcessElementFailure" are defined under "EquipmentFailure".

For "ProductFailure" and "ProcessFailure", the "Condition" class represents the conditions that must be satisfied. Fig. 3 shows a part of the subclasses of "Condition". In addition, the following two properties are defined to represent a failure as a state in which a certain condition is impaired in a certain process. In this paper, a property is represented as "property name (domain class, range class)".

- fail_Condition(State, Condition): A relationship between a failure and a condition impaired. In Fig. 4, "assembly misalignment" fail_Condition "parameter condition".
- happen_in_Action(State, Action): A relationship between a failure and a process or an action where it happens. In Fig. 4, "assembly misalignment" happen_in_Action "assemble".

Fig. 5 is a part of the subclasses of "PhysicalFactor" and "HumanFactor", which are the subclasses of "ProcessElementFailure".

D. Inference

1) Inference Overview: The framework receives the following two inputs.

- Text about the failure that occurred on the target system

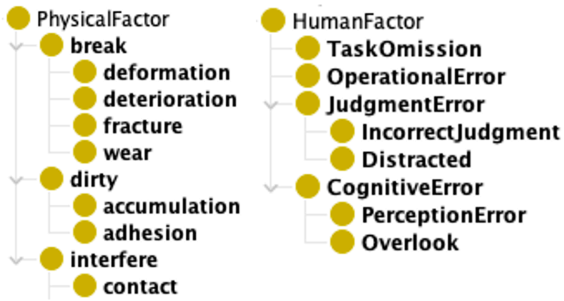

Fig. 5. "ProcessElementFailure" has "PhysicalFactor" and "HumanFactor" as its subclasses.

- The position where the failure occurred on the process order model of the target system

The following procedure is performed to infer candidate failure causes for the input.

1) Calculate the similarity between the input failure and the failure descriptions in the database.
2) Extract the node of the process order model with the highest similarity to the original function of the candidate cause.
3) Generate candidate causes that can occur near the process order model node.
4) Output the candidate causes in order of similarity.

The similarity calculation method used in this framework is that of Okazaki et al[8]. The similarity between instances of the past cases ontology is calculated based on the similarity between classes in the domain ontology.

2) Inference Details: After generating the ontology instance for the input failure, the similarity with each instance of "FailureElement" in the past cases ontology is calculated. Let $Sim_{Failure}$ denote the similarity calculated here. For each "FailureElement" in the range of has_Cause, the node of the process order model with the highest similarity to the "Function" connected to the "FailureElement" by has_Function is extracted, and let $Sim_{Function}$ denote the similarity calculated here. Based on each "FailureElement", a candidate cause is generated so that it has the highest similarity to the "FailureElement" and is near the node of the process order model. If there is a "FailureElement" that does not have a "Function", the process order model is used to narrow down a reasonable failure cause candidate. The overall similarity Sim_{output} is calculated as below and the candidate causes are output in order of Sim_{output}.

$$Sim_{output} = Sim_{Failure} \cdot Sim_{Function} \tag{1}$$

III. EXPERIMENT

The purpose of the experiment in this study is to verify the effectiveness of using maintenance logs and failure ontology with the proposed method in failure cause inference of manufacturing systems. The experiment consists of the following two parts.

1) To examine the effect of describing functions in maintenance logs. The following two inferences are compared.
   1-1) Inference using maintenance logs without descriptions of functions
   1-2) Inference using maintenance logs with descriptions of functions
2) To examine the effect of the entire proposed method. The following three inferences are compared.
   2-1) Inference using FMEA only
   2-2) Inference using FMEA and maintenance logs
   2-3) Inference using the failure ontology in addition to FMEA and maintenance logs

In the experiments, FMEA and maintenance logs on manufacturing systems similar to the inference target system are used. The target system is a demonstration system of a chip assembly line in an automobile parts manufacturer. The system consists of processes such as circuit board assembly, adhesive application, chip assembly, and surface visual inspection. The FMEA database contains 93 items for a pressure sensor assembly line in the same manufacturer. As maintenance logs, 89 items were used, which were rewritten into the proposed format from the candidate failure causes listed by experts in Okazaki et al.'s study[8] on a part of the LEGO car assembly line.

Two assumed failures are set on the target system: (a) misalignment of circuit board assembly, (b) chucking error of circuit board.

A. Construction of Inference Framework

Using Gaphor software for easy drawing of SysML diagrams, 19 activity diagrams, and 13 state machine diagrams are described for the target system. By defining them as a partial-order set, the process order model consisting of 141 nodes was generated.

As in Okazaki et al.[8], the ontologies were constructed using GiNZA and KNP as Japanese natural language processing tools, and Owlready2 as a module for handling ontologies with Python3.

B. Evaluation Method

In the experiments, the outputs of the inferences were evaluated by comparing them to the candidate failure causes listed by experts for the assumed failures. This evaluation method is chosen because the goal of inference is to output the list of failure causes that is identical to the list enumerated by the experts. The experts were two people with decades of experience in designing and setting up assembly systems. In the interviews, they listed the possible failure causes for the two assumed failures based on a video of the target system in operation and detailed illustrations of the process around where the assumed failures occurred. In this study, the candidate failure causes they listed are called "correct answers". We obtained 19 correct answers for Failure (a) and 13 correct answers for Failure (b). For the x-th output, the following two metrics are calculated.

- precision: the ratio of outputs that match the correct answers among the first to x-th outputs
- recall: the ratio of correct answers covered by the first to x-th outputs

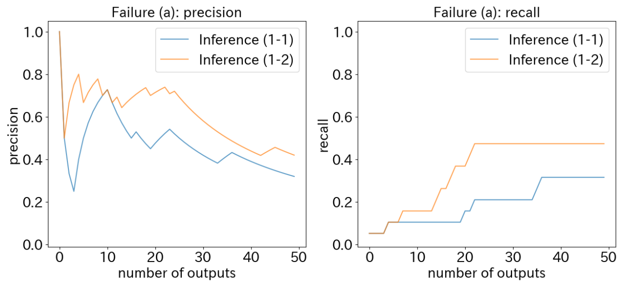

Fig. 6. Evaluation of using maintenance logs with descriptions of functions: Failure (a)

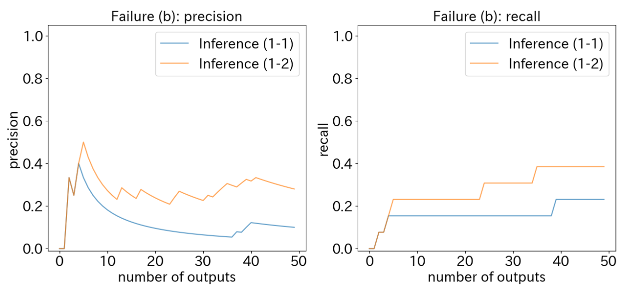

Fig. 7. Evaluation of using maintenance logs with descriptions of functions: Failure (b)

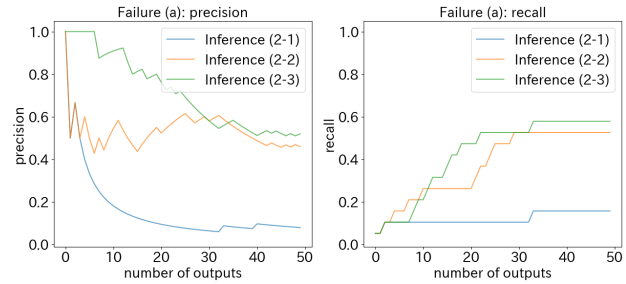

Fig. 8. Evaluation of the entire proposed method: Failure (a)

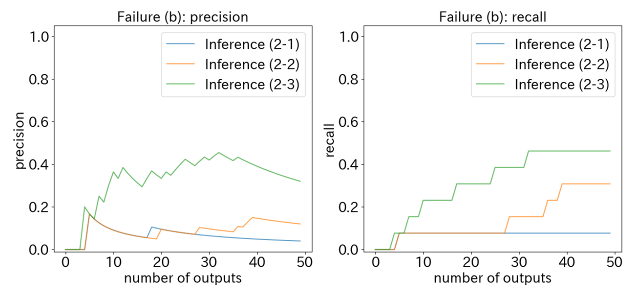

Fig. 9. Evaluation of the entire proposed method: Failure (b)

Considering that an investigation of a failure cause starts from the earliest output in practice, evaluations were conducted for the first 50 outputs. For practical use, this experiment aimed for a recall of 0.7 for 50 outputs.

IV. RESULTS & DISCUSSION

A. Experiment on the Format of Maintenance Logs

Fig. 6 is the result of Failure (a) and Fig. 7 is the result of Failure (b). In both cases, Inference (2-2) using maintenance logs with descriptions of functions showed higher values for both precision and recall in all ranges of output order. Inference (1-2) had a recall of 0.48 for Failure (a) and 0.38 for Failure (b) for the 50 outputs, 1.5 times and 1.7 times higher than Inference (1-1), respectively.

One of the reasons for this is that the description of functions in maintenance logs makes it possible to compare the target system with the system which the maintenance logs are about. In the inferences, information about the target system is available in the framework as the process order model, while information about the manufacturing systems which the maintenance logs are about is not stored. Without descriptions of functions in the maintenance logs, it is impossible to compare the similarities between the systems, and inference is limited to extracting similar failures from the maintenance logs and replacing them with possible failure causes in the target system. On the other hand, when maintenance logs have descriptions of functions, the process order model can be used to narrow down the candidates of failure causes based on the fact that failure causes and their effects are propagated from upstream to downstream of the process in manufacturing systems. For this reason, it is considered that the maintenance logs with descriptions of functions can output more valid failure cause candidates.

As a secondary effect, the domain ontology can be improved by simply increasing the information about manufacturing systems in the maintenance logs, thereby improving the quality of the inference. The domain ontology, except for the failure ontology constructed in advance from the knowledge of experts, covers a narrow domain, mainly the FMEA database and the maintenance log database. Therefore, the increase in the number of descriptions in maintenance logs is supposed to have a significant impact on the richness of the domain ontology. After all, these are synergistic effects, and it can be said that maintenance logs with descriptions of functions are effective in improving the quality of failure cause inference in manufacturing systems.

B. Experiment on the Entire Proposed Method

Fig. 8 is the result of Failure(a) and Fig. 9 is the result of Failure(b). For both precision and recall, Inference (2-2) using FMEA and maintenance logs was better than Inference (2-1) using FMEA alone, and Inference (2-3) with the addition of the failure ontology tended to be the best. The recall of Inference (2-3) with 50 outputs was 0.58 for Failure (a) and 0.45 for Failure (b), which were 3.7 and 5.0 times higher than those of Inference (2-1), respectively. Considering this, the entire proposed method proves to be effective in improving the quality of inference.

The effect of using the failure ontology to improve quality was particularly significant in Failure (b). The characteristics of the terms are considered to be the reason for this. While the term "assembly misalignment" was included in the FMEA and maintenance logs used, "chucking error" was not, and "chucking error" is a vague term that refers to various

TABLE II
EXAMPLES OF DUPLICATED OUTPUTS

Failure Cause	Causal Factor
[3-axis robot, descent], misalignment	
3-axis robot, descent misalignment	
palette, stopping position misalignment	stopper, wear
palette, position misalignment	stopper, wear

situations such as chuck misalignment, chuck timing error, and workpiece drop during chucking. That is why extracting similar failures was difficult without a failure ontology, while it became possible by using the failure ontology to describe the process in which chucking error occurs and the condition impaired.

However, Inference (2-2) without the failure ontology showed higher recall than Inference (2-3) with the failure ontology in the range up to about 10 outputs. Since the precision is higher for Inference (2-3), it can be assumed that the improvement in recall was prevented by the duplicates of outputting identical or similar cause candidates because recall is a measure of comprehensiveness. Looking at the actual output, it can be confirmed that some outputs are not exactly the same but have the same meanings, as shown in Table II. The following reasons can be considered for the duplicates.

- The databases contain several similar past cases, which generates outputs of similar failure cause candidates.
- Some of the properties defined in the domain ontology have multiple classes as domains and ranges, therefore, multiple patterns are possible when generating output.

Since investigations are prioritized based on the order of output in practice, the inference outputs within the early order are extremely important. Therefore the identification and handling of duplicated outputs remain challenges for future works.

Even with the improved inference quality by the proposed method, the recall was at most about 0.6, which is a low result overall. Looking at specific outputs, there were several outputs such as "stopper unreadable" that had word combinations that were never possible in manufacturing systems. To reduce such outputs, it is important to express the differences between concepts in manufacturing systems more clearly, therefore, it is necessary to subdivide the classes and properties of the domain ontology. Another possible reason for the overall low recall is the use of the model that focuses only on the process order of the target system. A model representing the hierarchical structure of processes and actions, not only process order, should be used, for example, when examining an assembly process in detail, it includes chucking, carrying, and press fitting. Such a model should be used to select the granularity of descriptions according to FMEA and maintenance logs.

V. CONCLUSIONS

In this study, we proposed a description method for maintenance logs to achieve high reusability and a failure ontology based on expert knowledge. We utilized these methods to infer the failure causes of a manufacturing system using FMEA and maintenance logs from systems similar to the target system. The proposed method showed higher precision and recall than using FMEA alone, and the recall of 50 outputs was 3.7 to 5.0 times higher than that of FMEA alone. This suggests that the proposed methods are effective in inferring failure causes.

As a prospect, we aim to enhance inference quality by constructing a model that can represent not only the process order of the target system but also the hierarchical structure of processes and actions. Additionally, to further enhance the quality of inference, it is also important to identify and handle duplicate outputs. The occurrence of duplicate outputs is attributed to the presence of multiple similar descriptions in past cases, which indicates a high frequency of input failures caused by the duplicated failure causes. This suggests that the duplicates could be effectively utilized. Furthermore, in the future, we will verify the extent to which this method can be generalized across different types of manufacturing systems, including larger-scale and more complex ones.

REFERENCES

[1] Z. Wu, W. Liu, and W. Nie, "Literature review and prospect of the development and application of FMEA in manufacturing industry," *The International Journal of Advanced Manufacturing Technology*, vol. 112, no. 5-6, pp. 1409 − 1436, Jan. 2021.

[2] L. Jean-Baptiste, "Ontologies with Python: Programming OWL 2.0 Ontologies with Python and Owlready2," Apress, 2021.

[3] W. L. Mikos, J. C. E. Ferreira, P. E. A. Botura, and L. S. Freitas, "A system for distributed sharing and reuse of design and manufacturing knowledge in the PFMEA domain using a description logics-based ontology," *Journal of Manufacturing Systems*, vol. 30, no. 3, pp. 133 − 143, Aug. 2011.

[4] A. Zhou, D. Yu, and W. Zhang, "A research on intelligent fault diagnosis of wind turbines based on ontology and FMECA," *Advanced Engineering Informatics*, vol. 29, no. 1, pp. 115 − 125, Jan. 2015.

[5] Z. Rehman and C. V. Kifor, "An Ontology to Support Semantic Management of FMEA Knowledge," *International Journal of Computers Communications & Control*, vol. 11, no. 4, p. 507-521, Jul. 2016.

[6] A. Aamodt, E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *AI Communications*, vol. 7, no. 1, pp. 39-59, 1994.

[7] W. L. Mikos, J. C. E. Ferreira, and F. G. C. Gomes, "A distributed system for rapid determination of nonconformance causes and solutions for the thermoplastic injection molding process: A Case-Based Reasoning Agents approach," *2011 IEEE International Conference on Automation Science and Engineering*, pp. 755 − 760, Aug. 2011.

[8] S. Okazaki, S. Shirafuji, T. Yasui, and J. Ota, "A Framework to Support Failure Cause Identification in Manufacturing Systems through Generalization of Past FMEAs," *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 859-865, 2023.

[9] M. Devaney, A. Ram, H. Qiu, and J. Lee, "Preventing failures by mining maintenance logs with case-based reasoning," *Proceedings of the 59th meeting of the society for machinery failure prevention technology (MFPT-59)*, 2005.

[10] S. Friedenthal, A. Moore, and R. Steiner, "A Practical Guide to SysML," Morgan Kaufmann, 2009.