2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

A Framework to Support Failure Cause Identification in Manufacturing Systems through Generalization of Past FMEAs

Sho Okazaki[1], Shouhei Shirafuji[2], Toshinori Yasui[3], and Jun Ota[2]

Abstract— **This study proposes a framework for inferring the causes of failures occurring in manufacturing systems from past Failure Mode and Effect Analyses (FMEAs) conducted on other systems to assist in inspecting and maintaining the systems. Among various manufacturing systems, a framework to search past FMEAs and the corresponding causes of the failure requires solving the following problems. First, the difference in products, equipment, and wording to represent them make it difficult to search the similar failure phenomenon from FMEAs. Secondly, the causes of failure highly depend on the process flow of the system until the failure occurs. Therefore, it is also hard to find appropriate failure causes from FMEAs without reflecting on the process. The framework solves the first issue by generalizing descriptions in past FMEAs based on structured concepts of manufacturing systems in an ontology before inference of causes to address. Furthermore, the framework analyzes the correspondence of the process flows between the target manufacturing system and past FMEAs using a process order model generated by SysML diagrams to solve the second issue. The comparison between the causes inferred by the proposed framework and by skilled experts for three typical failures in the manufacturing system and the interview with them about the plausibility of the inference results showed that more than 73 % of them were valid.**

I. INTRODUCTION

Inspection and maintenance of manufacturing systems require experts who are familiar with the system's structure and potential defects that may occur. It is a concern in the Japanese manufacturing industry that the shortage of experts makes identifying the defect causes and maintenance activities difficult in the future. A practical approach to compensate for the lack of engineering skill is to refer to the past failure analysis that experts have conducted to identify the causes of failures and repair them. One of the most available data in the factory about failure analysis is the past failure mode and effect analysis (FMEA), which is widely used in machinery, electronics, chemicals, pharmaceuticals, and textiles to improve system quality and reliability [1]. FMEA is a typical method for analyzing the causes and effects of potential failures and a qualitative reliability analysis method that predicts all potential failure modes and analyzes their causes and effects in advance by analyzing aspects such as the system's structure, features, and functions by a team of mainly experts. Therefore, the database of past FMEAs contains experts' knowledge in failure analysis, from specified for an individual system to general for inspection and maintenance of the system. However, it is difficult to find descriptions of similar failures in the database of FMEAs for manufacturing systems, which differs from the manufacturing system in question due to differences in products, equipment, process, and wording. To overcome the low reusability of existing FMEAs and use them efficiently, it is essential to develop a framework that supports cause identification by retrieving relevant knowledge from FMEAs for failures in a manufacturing system.

In recent years, diagnostic systems using previous knowledge have attracted much attention in industries [2], [3], [4]. Mikos et al. [5] and Rehman et al. [6] used ontologies and SPARQL (Simple Protocol and RDF Query Language) to reuse FMEAs. Ontology is a formal model of the structure of a domain by organizing entities into concepts and relationships, where a concrete element is represented as an instance of its corresponding concept [7], [8]. SPARQL is a query language for graph data, such as ontologies [9]. These studies constructed ontologies for FMEA to organize FMEA's contents and queried the ontologies with SPARQL to search for failure-related items in the FMEA database. Their framework to store past FMEAs in an ontology and search for the causes of failures by SPARQL queries made it possible to assist in identifying the causes of failures. In addition, Zhou et al. [10], Mikos et al. [11], and Camarillo et al. [12] used case-based reasoning (CBR) to realize a more flexible search of the FMEA database. They used the nearest neighbor method to find similar past cases from the FMEAs based on the coincidence rate of defined characteristics between the failures, such as the attribute of the failure, product part number, and frequency of occurrence.

However, to flexibly reuse failure-cause relationships commonly appearing in FMEAs for various systems, the generalization of their descriptions is necessary to address the following issues. First, differences in components (e.g., equipments and materials) among manufacturing systems complicate the identification of similar failure phenomena in their respective FMEAs. Furthermore, variations in wording used by different engineers further complicate the issue. Secondly, to identify similar failure phenomena, similarities not just in components and wording, but also in the system's process need to be taken into account. Failures typically propagate in the system as reflected in the failure, cause, and effect in FMEAs. Therefore, evaluating the similarities between different manufacturing systems and failures in

*This work was supported by New Energy and Industrial Technology Development Organization (NEDO)

[1]Sho Okazaki is with Department of Precision Engineering, School of Engineering, The University of Tokyo, 113-8656, Tokyo, Japan

[2]Shouhei Shirafuji and Jun Ota are with Research into Artifacts, Center for Engineering School of Engineering, The University of Tokyo, 113-8656, Tokyo, Japan

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

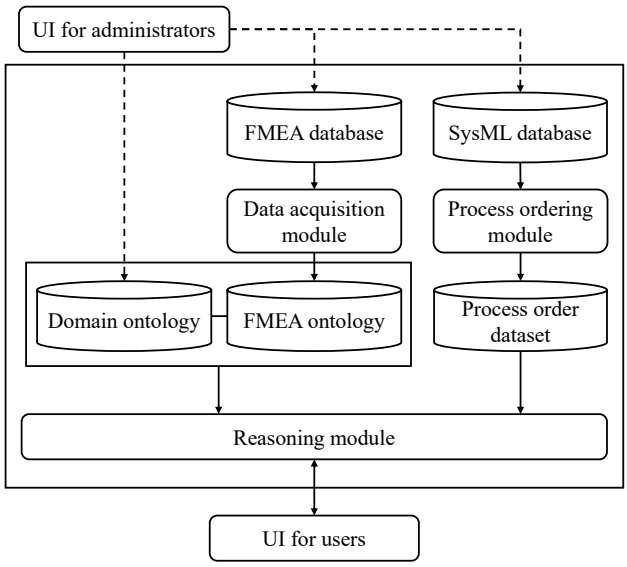

Fig. 1. Overview of the proposed framework

these systems is essential to search for similar failures in the past FMEAs for failure analysis in the launching or maintenance of the facility.

This study proposes a framework for reasoning the cause of failure in a manufacturing system based on past FMEAs conducted for other manufacturing systems from the similarity of the terms and process. The framework realizes the organization of the concepts in the manufacturing system to generalize FMEAs with different components, including equipment and materials, and with different wording by constructing a domain ontology, which defines and structures the concepts of the specific domain and the relationships between concepts. Furthermore, the framework narrows down the corresponding past FMEAs reflecting the process flow of the manufacturing system. The framework constructs this process flow by defining sequential flows among information and goods in the processes of a manufacturing system described by SysML (System Modeling Language). SysML is a modeling language developed for engineering systems based on UML (Unified Modeling Language) and used to describe complicated systems, such as manufacturing systems, including their behaviour and state [13]. In the following section, we introduce the method used in the framework, its implementation, and the evaluation through the experiment to compare the reasoning results using the framework with skilled experts' diagnoses.

II. METHOD

Fig. 1 illustrates the framework for identifying causes of failures in manufacturing systems. The user inputs a failure description and process location where the failure occurs through the UI and receives candidate causes inferred from existing FMEAs as outputs.

The FMEA ontology categorizes the FMEA data into four concepts: process function, failure mode, failure cause, and failure effect, and maintains the relationships of the

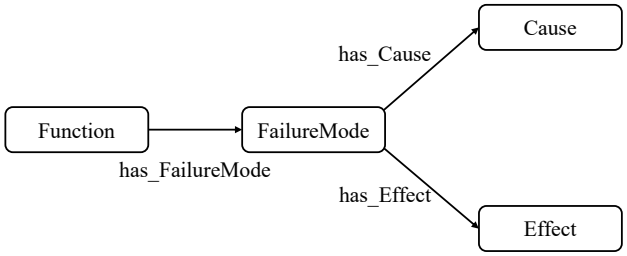

Fig. 2. FMEA ontology

concepts in each FMEA description. The FMEA ontology allows FMEA descriptions to be managed as unique elements (instances) of each concept while preserving the relationships among the elements. The domain ontology structures the concepts in a manufacturing system, such as action, equipment, material, parameter, and state, and defines the relationships among the concepts. The domain ontology allows the concepts of a manufacturing system within the description of an instance of FMEA to be as concepts and their relationships in a domain ontology. The process order dataset manages the sequential relationship of process flows defined in the SysML diagrams of the manufacturing system processes. The process order dataset allows us to carry out the correspondence of the spread of cause-effect relationships and the cause-and-effect of failure phenomena and narrow it down to relevant cases based on knowledge of past failures about the relevant manufacturing system.

The reasoning module inferences the cause of failure. The domain ontology generalizes the descriptions in the FMEA ontology to infer problems similar to previous FMEAs. The process order dataset narrows the inference of possible causes in the system by considering the process flow. The below sections describe three essential processes in the framework in more detail. This framework was constructed with FMEAs and SysML diagrams written in Japanese and translated into English for better understanding in the following sections.

A. Retrieval of Cause Candidates using FMEA Ontology

1) Construction of FMEA Ontology: FMEA analyzes the causes and effects of all potential failure modes for the elements of a process. In order to reuse the content described in FMEA, it is practical to structure the relationship between each element as an ontology. Here, based on the fact that FMEA often lists failure modes for process functions and analyses the causes and effects of the failure modes, the FMEA ontology consists of four elements: "Function (process function)", "failure mode (failure mode)", "cause (failure cause)", and "effect (failure effect)", as shown in Fig. 2. In addition, the FMEA ontology defines the relationship between classes; each process has failure modes (*has_FailureMode*), and the failure modes have the causes and effects (*has_Cause*, *has_Effect*). This process classifies the elements described in the FMEA data into instances of each class. The proposed framework uses this FMEA ontology to structure the FMEA description and express the

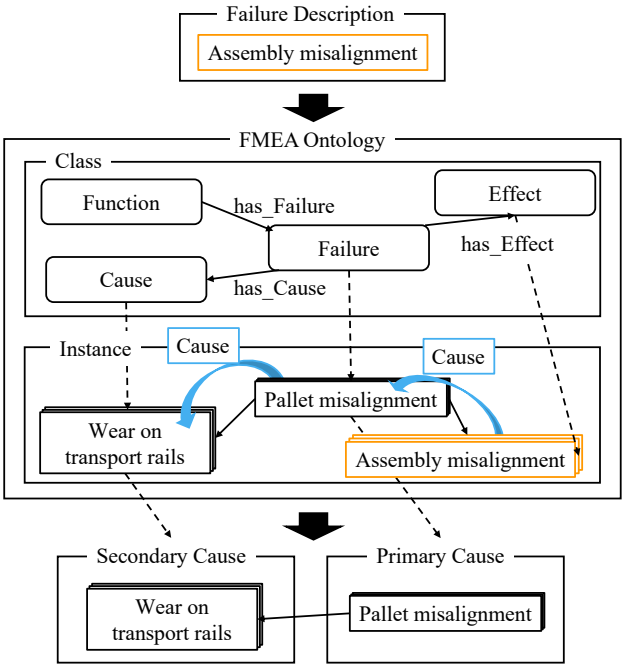

Fig. 3. Example: For each failure description in the input, the "Effect" of the corresponding FMEA ontology is extracted and the "Failure" and "Cause" for it are extracted and output as the primary and secondary cause, respectively.

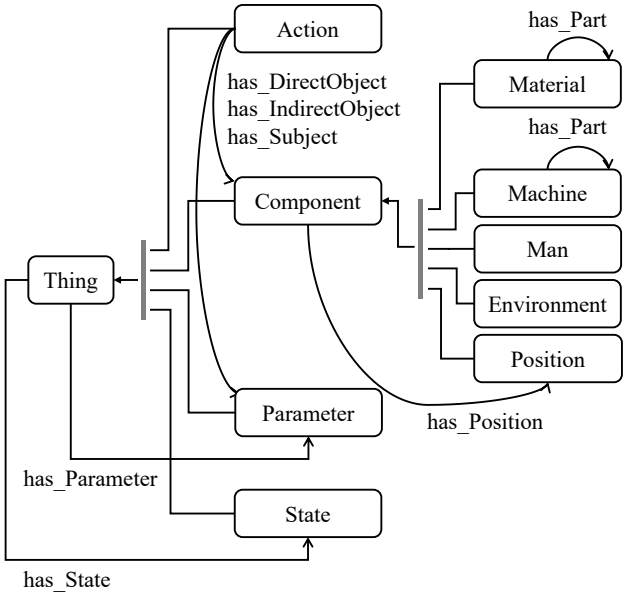

Fig. 4. Overview of the domain ontology

causal relationship between failure modes and their causes and effects based on the process function.

2) Retrieval of Failure Cause Candidates: This process retrieves the elements associated with the failure described in the input using the FMEA ontology. Furthermore, this process extracts "FailureMode" and "Cause" for "Effect" in the failure description as a primary and secondary cause for resultant failure, respectively, linking them to "Function". Fig. 3 shows an example of the retrieval of candidate failure causes. First, all instances in "Failure" associated with "Effect" instances corresponding to the failure description (ex. "Assembly misalignment") are extracted (ex. "Pallet misalignment"). Secondly, the framework extracted all instances in "Cause" associating with the extracted "Failure" instances (ex. "Wear on transport rails"). Finally, these instances are obtained as primary and secondary causes while maintaining the relationship between them.

B. Generalisation of Descriptions using Domain Ontology

1) Construction of Domain Ontology: This process constructs the domain ontology by organizing the concepts specified for manufacturing systems to describe process functions, failure modes, failure causes, and failure effects in FMEA. First, the classes are defined as shown in Fig. 4, taking into account the semantic structure of the description. In the domain ontology, "Thing" represents the top-level concept of the entire domain. The lower-level concepts of "Thing" are "Action" for the actions in the process, "Component" for the components of the process, "State" for the states of other concepts, and "Parameter" for the parameters of other concepts. "Component" is also classified into "Man", which represents a human being, "Material", which represents parts and materials, "Machine", which represents equipment, "Environment", which represents disturbance around the system, and "Position", which represents a part or position of an object. By creating a hierarchy of superordinate and subordinate relationships according to the level of abstraction, it is possible to manage the different levels of abstraction of the words described in the FMEA.

Secondly, those classes have the properties defined as follows. In this paper, a property is represented as a property name (domain class, range class). In manufacturing systems, the situation differs significantly between direct and indirect objects and also differs greatly depending on whether the subject is equipment or a human being. Therefore, the class associated with "Action" have three categories broadly classified: direct object, indirect object, and subject, and their definitions are *has_DirectObject*(*Action*, [*Component*, *Parameter*]), *has_IndirectObject*(*Action*, [*Component*, *Parameter*]), and *has_Subject*(*Action*, [*Man*, *Machine*]), respectively. Furthermore, classes in manufacturing systems have states and parameters defined as *has_State*(*Thing*, *State*) and *has_Parameter*(*Thing*, *Parameter*), respectively. In addition, each piece of equipment and material has a part defined as *has_Part*([*Material*, *Machine*], [*Material*, *Machine*]). Finally, the situation differs depending on the position of the process element, and it is defined as *has_Position*(*Component*, *Position*).

2) Representation of FMEA Descriptions: In inferring the failure cause, the domain ontology generalizes the representation of the FMEA ontology instances. First, the framework applies morphological analysis and dependency structure analysis to the descriptions in FMEA to obtain the classes and their dependencies. Next, the framework

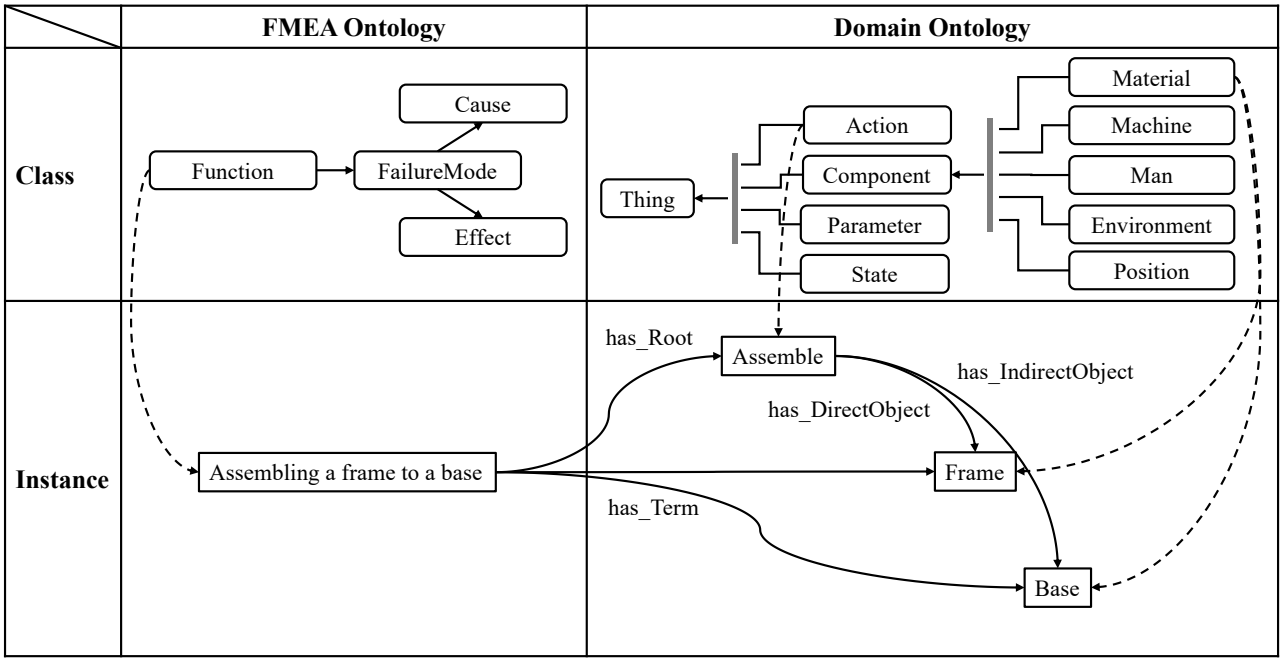

Fig. 5. Example: An FMEA description, represented as an instance in the FMEA ontology, is mapped to instances and properties in the domain ontology.

defines the classes in the domain ontology and the properties between them according to the rules created from the type of dependencies. For example, if there is a direct object dependency between *Action* and *Component*, the relationship is defined as *has_DirectObject*. Then, the FMEA ontology's description becomes the domain ontology's instances and their properties.

Fig. 5 shows an example of the representation of an FMEA description. "Function" of "Assembling a frame to a base" in an instance is composed of the concepts of materials, "Frame" and "Base" and an action, "Assembly". They have the relationship that "frame" is the direct object of "assembly" and "base" is the indirect object of "assembly".

3) Generalisation of FMEA Descriptions: In Section II-A, the reasoning module of the framework calculates the similarity between the failure descriptions in the input and the description of "effects" in the FMEA to extract the FMEAs that describe similar failure. It calculates similarity with concepts and properties of them in the domain ontology. Furthermore, the reasoning module also calculates the similarity between the process behavior described in SysML of the target system and the "Function" in FMEA to estimate the corresponding behavior of the process. Then, the reasoning module determines possible causes by evaluating if they are included in the corresponding SysML description.

Fig. 6 shows an example of the generalization of FMEA descriptions. In the example, the module generalized the cause description "Wear on transfer rails". As a result, it extracted the similar classes "Belt conveyor", "Screw conveyor", and "Transfer rail" by calculating the similarity from the hierarchical relationship between the classes and their properties, that is "Wear on transfer rails" may transformable into "Wear on belt conveyor" and "Wear on screw conveyor".

C. Narrowing Down Candidate Causes using the Process Order Model

1) Process Order Dataset: The process order model in the dataset is a model that describes the order relationship between the elements that possibly cause and possibly affect failures in a manufacturing system. The framework generates a model defining and describing each element of a manufacturing system as a partially ordered set for given diagrams representing behavior in a manufacturing system. The framework creates the process order model from the diagrams used to express the state and behavior of the system, such as activity diagrams and state machine diagrams described in SysML, by analyzing the names and definitions in the diagrams. Diagrams in SysML represent the relationships of information and goods as the token flow in an activity diagram and state transition in a state machine diagram. First, the framework generates the local process order model defining the order relationships between the elements in each diagram, focusing on the flow of information and goods. After that, it generates the order relationships of the entire manufacturing system by defining the order relationships between the diagrams based on the description in the diagrams as follows: When a state in a state machine diagram triggers an action in an activity diagram, the state is greater than the action in the partial-order set of the process order model. When an action in an activity diagram triggers a state transition in a state machine diagram, the action is greater than the state in the partial-order set of the process order model.

2) Narrowing Down Candidate Causes: The above process order model, which defines sequential relationships of information and goods in the process of a manufacturing sys-

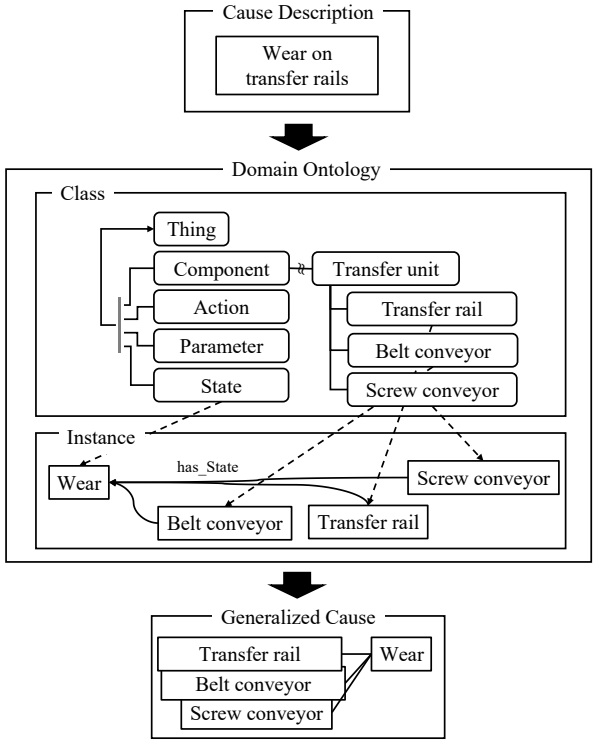

Fig. 6. Example: The cause description extracted from the FMEA ontology and represented by the domain ontology is replaced by similar concepts and generalized focusing on the similarity of the classes and the properties of the domain ontology.

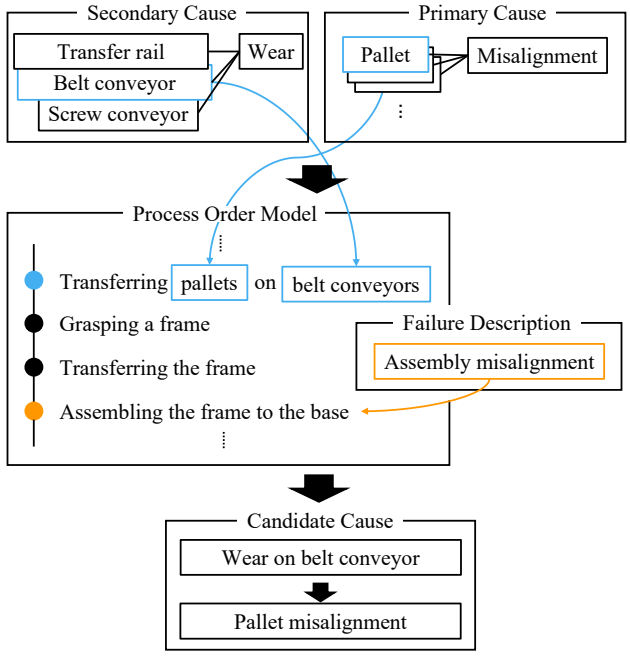

Fig. 7. Example: Mapping the classes in the generalized cause description to the process order model determines the appropriate class and narrows down the general failure causes to output primary and secondary causes.

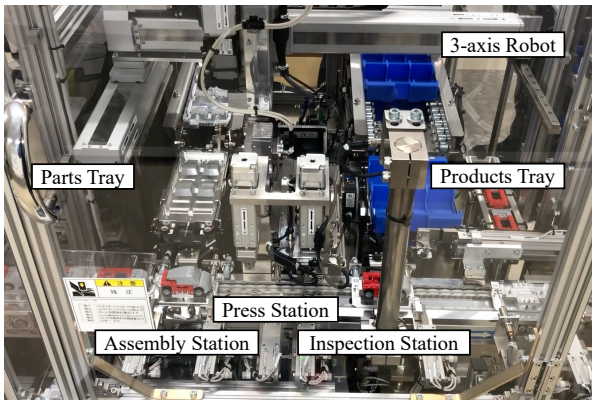

Fig. 8. In the target manufacturing system, using a 3-axis robot, a part on a parts tray are assembled onto the base of a Lego car, and then pressed from above. After that, height and image inspections are conducted, and it is transported to the products tray.

tem, allows us to evaluate whether the primary and secondary cause candidates associated with the model's behavior can occur in the manufacturing system. The framework represents descriptions of SysML as the classes and properties in the domain ontology constructed for FMEAs and assesses whether the classes of the SysML description contain classes of the generalized cause. It enables the framework to assess whether the cause can occur in the behavior of the process order model. In addition, a manufacturing system propagates the cause of failures from upstream to downstream. Therefore, the secondary cause must occur in the same or upstream process as the primary cause, and both need to be upstream of the failure phenomenon. Furthermore, the framework identifies a plausible cause candidate by assessing how closer the target of failure analysis, which corresponds to the "Function" in FMEA, is to the cause in the process order model. Then, the framework narrows down the candidate causes to the causes that can occur in the target manufacturing system, taking into account the order of the processes.

Fig. 7 shows an example of narrowing down candidate causes using the process order model in the framework. First, the framework checks whether the classes of the domain ontology in the generalized secondary and primary causes exist in the classes of the description on the process order model. For example, there are "Transfer rail", "Belt conveyor" and "Screw conveyor" in the Secondary cause, but only "Belt conveyor" exist in the description of the process order model as part of the description "transferring pallets on belt conveyors". Since the process order model does not include the other classes, the framework narrows down the candidate class only to "Belt conveyor". In the example, the framework similarly narrowed down the primary cause into "pallets are misaligned due to wear on the belt conveyor".

III. EXPERIMENT

In the experiment, we compared the failure causes inferred by the proposed framework through the past FMEAs with the causes anticipated by experts for possible failure phenomena in a manufacturing system to evaluate the effectiveness and clarify the proposed framework's problems. For this experi-

ment, we prepared FMEA data analyzed for a manufacturing system as the basis of the inference framework and another manufacturing system as the cause inference target. The former manufacturing system for which the FMEA data was created is a part of the process of a pressure sensor assembly system. This system bonds IC chips, assembles them into a case to make a pressure sensor, and inspects them. The latter manufacturing system for the causal inference testing of given failures is a part of the process of the automated assembly system for Lego brick cars, as shown in Fig. 8. This system assembles a miniature Lego brick car using air cylinders and a three-axis robot with some inspection processes. First, we constructed the ontology based on the above FMEA data and prepared SysML diagrams for the LEGO car line. Second, we prepared failure phenomena that can cause in the LEGO car line and infer their cause using our framework. Prepared failure phenomena are three representative failures: (a) “assembly misalignment”, (b) “misjudgment in image inspection”, and (c) “error in QR reading”. Finally, through interviews with two experts, we compared the candidate failure causes listed by them with those output by the proposed framework to evaluate the validity of the candidate causes output by the proposed framework. The followings are the framework implementation details, the data we used in the experiment, and its procedure.

A. SysML

We described the SysML diagrams for the system of causal inference targets and managed them using Gaphor. Gaphor is software designed for SysML and UML drawing and programming management of drawn diagrams. Fig. 9 shows the Gaphor’s interface for SysML diagrams and the LEGO car assembling system diagrams. In the figure, Gaphor displays the activity diagram of the visual inspection of the Lego car assembly system in the center. On the left side, the items of each diagram are displayed, which allows comprehensive management of all diagrams for a manufacturing system. Fig. 10 shows the process order model for the LEGO car assembling system obtained from the diagrams in Gaphor through our framework. There are 17 activity diagrams and seven state machine diagrams for the Lego car assembly system. The framework structured the 91 elements in the SysML diagrams as a partially ordered set.

B. FMEA Data Acquisition

We adapted GiNZA and KNP in the FMEA data acquisition module of the framework. GiNZA is a natural language processing module for analyzing Japanese using spaCy, a fast and lightweight natural language processing framework, and sudachi, a morphological analyzer. KNP is a module that can also perform dependency structure analysis of Japanese. Although detecting parallel structures in Japanese descriptions is challenging, KNP improves analysis accuracy by analyzing them based on dependency structures. In our implementation of the structured module, GiNZA analyzes dependency structure, and KNP analyzes the parallel structures and adds them before constructing ontology.

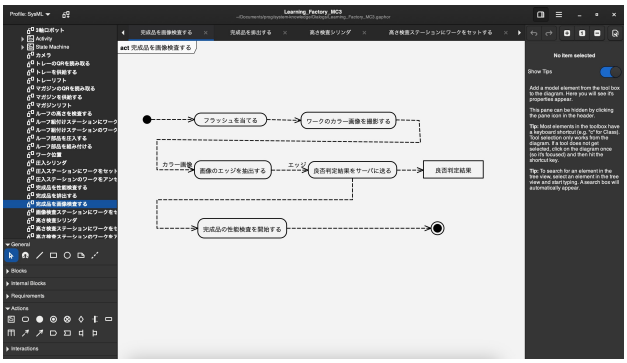

Fig. 9. The center of the Gophor’s interface displays a SysML diagram, and on the left side, various other diagram elements are available for editing.

Fig. 10. The Lego car assembly system’s process order model is comprised of the behaviors and states depicted in the SysML diagrams. Each element in the process flow is arranged from top to bottom in the diagram.

C. Construction of Ontology

Protégé and owlready2 are the software used for the ontology module of the framework. Protégé is the software used for ontology construction, which has an intuitive user interface for editing the ontology. Owlready2 is a module for the programming language Python, which allows the definition and editing of ontologies by programming. We used owlready2 to import the data in FMEA to protégé. Fig. 11 shows the protégé’s interface for the FMEA ontology managing the data generated from the FMEA of the pressure sensor assembling system. The left of the figure shows the classes in the FMEA ontology, and the bottom right shows the instances of “Effect”. The framework extracted the instances of the FMEA ontology from the FMEA worksheet described for the pressure sensor assembly system in the experiment. There were 56 *Function*, 63 *FailureMode*, 86 *Cause*, and 46 *Effect* instances. The framework used concepts contained in the description of the FMEA for the pressure sensor assembly system and that of the SysML diagrams for the Lego car assembly system to construct domain

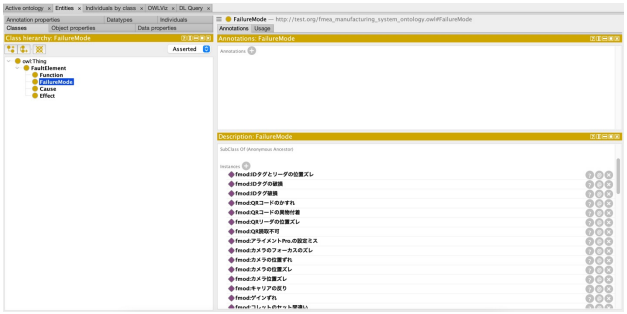

Fig. 11. The classes in the FMEA ontology are displayed on the left side of the Protégé interface, while the instances of "FailureMode" are located on the bottom right.

TABLE I
EVALUATION OF RESULT BY EXPERTS

	Failure (A)	Failure (B)	Failure (C)
Validity (%)	73	93.3	73.3
Comprehensiveness (%)	2.5	25	43.8

ontology. There were 69 *Action*, 106 *Component*, 179 *State*, 81 *Parameter* classes in the domain ontology.

D. Cause Inference

In reasoning, not only the order of the primary and secondary causes, but also the proximity of the *Function* was taken into account in narrowing down process. The cause candidates are narrowed down to those where (i) primary cause, (ii) secondary cause and (iii) *Function* are either (ii), (i), (iii) or (ii), (iii), (i) in the process order model.

To simplify, the reasoning module selects the most appropriate combination of primary cause, secondary cause, and *Function* from the FMEA ontology, and disregards other combinations

E. Interview and Evaluation

We interviewed two experts with decades of experience designing and setting up assembly systems to evaluate the cause reasoning plausibility. First, the experts read the document about the structure and process of the LEGO car assembly system to understand the failure phenomena and the situation around the place where the failures occur. Second, we asked them to answer the possible causes of the three types of failures in the system. After the interview, we compared the causes suggested by the framework with their estimated causes to check how many suggested causes were the same between the framework and the experts. Finally, we interviewed the experts again to review the causes that the framework suggested but not the same as the experts' suggestions to judge whether each was a possible cause.

IV. RESULT

For each of Failure (A), (B), and (C), the expert provided 40, 20, and 16 possible failure causes, respectively, and the framework provided 8, 15, and 17 candidate causes, respectively. Table I shows the evaluation of the results. The validity is the percentage of the framework outputs evaluated as valid failure causes by the experts. The result showed high validity for all problems, and the validity for the suggested causes for the three failures was 73, 93.3, and 73.3 %, respectively. Comprehensiveness is the percentage of the possible causes listed by the experts that match the framework's outputs. The result's comprehensiveness was different between the problems, and the comprehensiveness of the suggested causes for the three failures was 2.5, 25, and 43.8 %, respectively.

Table II shows some of the causes suggested by the framework and the corresponding past FMEA data used to infer in the framework. For example, for Failure (A), while one of the experts' answers was "Wear and dents in the magazine positioning area cause misalignment," the cause suggested by the framework was "The pallet is misaligned due to wear on the conveyor" (which we converted to natural language from the suggested causes in the table). The above results were the suggested causes that narrowed down the process using the process order model. For example, the narrowing down process excluded "the infrared sensor is displaced by the vibration of the lift" considering the order of the primary and secondary causes in the above example, because the lift lifts the pallet after the infrared sensor recognizes the frame in the process. Therefore, the narrowing down process excluded the nonfeasible causes where the downstream element of the process affected the upstream element based on the fact that the causality of a failure usually propagates from upstream to downstream.

The results showed that the generalization of the FMEA description allowed the framework to output valid causes for three types of failure, even though the two manufacturing systems do not have the same processes. In addition, considering the process flow in the manufacturing system using the process order model, the framework suggested possible causes excluding unlikely FMEA items through the narrowing down process. However, the proposed framework includes some failure causes that are not valid, some that cannot generally occur as events, and some that do not make sense to engineers. Ontology with the enhanced number of concepts and the relationships between them is essential to improve the framework to distinguish the systems' differences clearly.

The FMEA data used in the experiment only included the causes and effects of failure modes described comprehensively. In contrast, the experts focused on the specific parts that caused the failure from the viewpoint of actual manufacturing systems maintenance. This difference caused the result that the proposed framework suggested fewer causes than the experts' answers. Therefore, incorporating more detailed information, such as maintenance records, into its analysis will improve the framework. Furthermore, the framework can suggest more detailed causes of failure phenomena by generalizing it using not only FMEA but also maintenance records or using FMEA that can describe the situation in more detail.

TABLE II
A PART OF OUTPUT AND ITS FMEA DATA FOR FAILURE (A)

Output		FMEA		
Primary Cause	Secondary Cause	Function	Failure Mode	Cause
The pallet is misaligned	Wear on the conveyor belt	Transferring to the next process	The pallet is misaligned	Wear on the conveyor screw
Incorrect assembly conditions	Teaching misconfiguration (3-axis robot)	Assembling the chip	Incorrect assembly conditions	Program misconfiguration
Foreign objects on chucks (3-axis robot)	Foreign objects on chucks (3-axis robot)	Assembling the chip	Foreign objects on collets (for semiconductor chips)	Foreign objects on collets (for semiconductor chips)
Frames overloaded	Excessive transport speed	Transferring to the next process	Work overloaded	Excessive transfer speed

V. CONCLUSIONS

In this study, we proposed a framework for reasoning possible causes of failures in manufacturing systems based on the past FMEAs analyzed for various manufacturing systems. The framework generalizes past FMEA descriptions using a combination of classes and properties in the domain ontology of manufacturing systems. The framework searches the possible causes of given failure from the generalized FMEA descriptions through the narrowing down process to consider the possible cause that satisfies the process in the target manufacturing system represented by the partial-order model generated from SysML diagrams.

In the experiment, more the 73 % of failure causes suggested by the proposed framework were valid for all three failure phenomena, even though the manufacturing systems are different, including their items and facilities, between the FMEA data analyzed for and the target of the failure causes reasoning. Furthermore, 2.5 % to 43.8 % of the causes suggested by the framework coincided with the candidate failure causes listed by the experts. These results showed that the proposed framework effectively reuses the experts' knowledge included in past FMEAs by generalizing their descriptions. However, more than half of the proposed framework's causes did not coincide with the experts' answers due to the difference in the viewpoint between the engineer who described the failure causes in FMEA and the experts in the interview who mainly focused on manufacturing system maintenance. Establishing a framework that integrates maintenance records into FMEA data is necessary to get beyond these differences and make the suggested causes of the framework closer to the experts' answers.

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] B. Liu, J. Wu, L. Yao, and Z. Ding, "Ontology-based Fault Diagnosis," *Proceedings of the 11th International Conference on Computer Modeling and Simulation*, pp. 112–116, Jan. 2019.

[3] F. Xu, X. Liu, W. Chen, C. Zhou, and B. Cao, "Ontology-Based Method for Fault Diagnosis of Loaders," *Sensors*, vol. 18, no. 3, p. 729, Feb. 2018.

[4] M. Chen, R. Qu, and W. Fang, "Case-based reasoning system for fault diagnosis of aero-engines," *Expert Systems with Applications*, vol. 202, p. 117350, Sep. 2022.

[5] 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.

[6] 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.

[7] Steffen Staab and Rudi Studer, *Handbook on ontologies*. Berlin: Springer, 2009.

[8] G. L. Zúñiga, "Ontology," *Proceedings of the international conference on Formal Ontology in Information Systems - FOIS '01*, pp. 187–197, Oct. 2001.

[9] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF," *W3.org*, 2013. https://www.w3.org/TR/rdf-sparql-query/

[10] 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.

[11] 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.

[12] A. Camarillo, J. Ríos, and K.-D. Althoff, "Knowledge-based multi-agent system for manufacturing problem solving process in production plants," *Journal of Manufacturing Systems*, vol. 47, pp. 115–127, Apr. 2018.

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