Analysis of industrial product quality problems based on knowledge graph reasoning

Wei Zhai[a*], Minai Wu[b], Jianyong Feng[b], Ping Lee[b]
[a] Guangzhou Institute of Technology, Xidian University, Guangzhou 510555, Guangdong, China;
[b] Ningbo Shuaitelong Group Co., Ltd, Ningbo 315157, Zhejiang, China

ABSTRACT

The quality of products produced on the production line is particularly important for manufacturers. However, how to trace the cause of product quality problems is a common problem in the field of intelligent manufacturing, and has not received good attention and research. Solving this problem requires the person in charge of production to be very familiar with the production links in a specific production scenario and have good reasoning ability, which is often challenging and a key factor to the factory's production efficiency. In this paper, we designed a system for a manufacturer that uses the knowledge graph to reason and analyze the product quality in production. To solve the difficulty of tracing the cause of the problem in a specific production scenario, we carefully analyze and mine the original Manufacturing Execution System (MES) data set of that manufacturer, establish a knowledge graph model, and find out the shallow causes and deep correlations of quality problems through the custom rule reasoning of the Jena reasoning engine. The reasoning system overcomes the low accuracy and high time-consuming of traditional manual inspection and provides effective help for factories to improve production efficiency, save production costs, and reduce defective product rates.

Keywords: Industry 4.0, knowledge graph, Neo4j, microservice

1. INTRODUCTION

Recently, knowledge graph technology has been widely used in Internet fields such as search engines and intelligent question answering, and has also been initially applied in many fields. For example, in order to facilitate researchers to understand the pathogenic mechanism of COVID-19, a semi-automatic knowledge graph construction method uses a combination of crawler technology and manual methods to build a knowledge graph[1]. Reference[2] combined the BERT (Bidirectional Encoder Representation from Transformers) preprocessing model to design a method to collect financial knowledge graph data to complement the financial knowledge graph system. In Reference[3], for the field of e-commerce, a training model is proposed that provides services such as commodity classification and commodity recommendation for downstream tasks in the form of knowledge vectors. The above knowledge graph applications of varying degrees are concentrated in finance, e-commerce, medical and other industries. However, under the current background of Industry 4.0, the application of knowledge graphs in the field of intelligent manufacturing is still insufficient. First, the advantages and characteristics of knowledge graphs in the field of industrial manufacturing are not clear enough, and secondly, there is the problem of ambiguous industrial application scenarios[4]. Finally, establishing an appropriate knowledge graph model for a specific factory production scenario is the key to whether the entire application can function. For example, in the auto parts production scenario, when the products produced by the automated assembly line enter the quality inspection workshop, various quality problems are often found. For the product quality problems found during the quality inspection process, a manual investigation was required in the past. If the cause of the problem is identified, this process is accompanied by unfavourable factors such as reduced production efficiency, consumption of human resources, and inaccurate traceability of the cause, resulting in loss of factory benefits. In this paper, we study how to establish a knowledge graph model for the production environment under the above circumstances. Although the existing MES data in the factory has relatively complete production information, how do establish a knowledge inference based on these data? The graph model is still difficult, and the process of knowledge reasoning can be simply summarized as the process shown in Figure 1. Since product data and information in the production process are often isolated from each other, it is necessary to be very familiar with the business process before establishing a basic knowledge graph. In order to solve this problem, we first mine and analyze the MES data, and will build a four-layer model of the knowledge graph divided into product

[*] wzhai@stu.xidian.edu.cn

Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12506, 1250668
© 2022 SPIE · 0277-786X · doi: 10.1117/12.2662045

nodes, quality inspection problems that occur, direct causes, and underlying factors. Then, a knowledge reasoning model using Jena reasoning engine for custom rule reasoning is proposed, and a supporting APP software based on microservice architecture is developed to provide it to production managers. This will provide a useful reference for factories in the field of intelligent manufacturing with such problems.

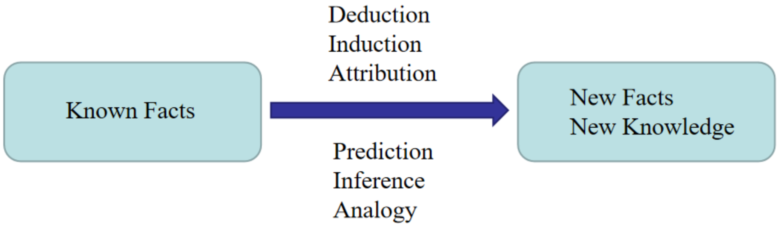


Figure 1. The process of reasoning.

The rest of this article is organized as follows. In Section 2, we briefly outline our solution. In Section 3, the knowledge graph model designed in this paper is introduced. Section 4 introduces the custom rule inference method based on Jena inference engine. In Section 5, the quality problem reasoning APP of the microservice architecture developed in this paper and its performance are introduced. Finally, Section 6 concludes.

2. OVERVIEW

In this chapter, we will outline our solution, introducing the system in the form of a flowchart. As shown in Figure 2, this system is based on Jena reasoning engine, uses Neo4j graph database as a storage medium, and conducts reasoning analysis for the product quality problems that occur in the assembly line production of an auto parts manufacturer. After exporting the CSV (Comma-Separated Values) file from Neo4j, we need to call the relevant functions in the Java toolkit to convert it into an RDF (Resource Description Framework) file, and generate the input data of the Jena inference engine for inference execution. The whole system is based on Java Spring Boot microservices, and uses Nginx and Nacos to manage and deploy microservices. It can be seen from the figure that when the reasoning engine obtains the relevant reasons for the product quality problem, it will be displayed on the front-end web page, and at the same time, the knowledge graph model in the original graph database will be updated and updated. Each time the reasoning analysis is carried out, the result will be more and more accurate.

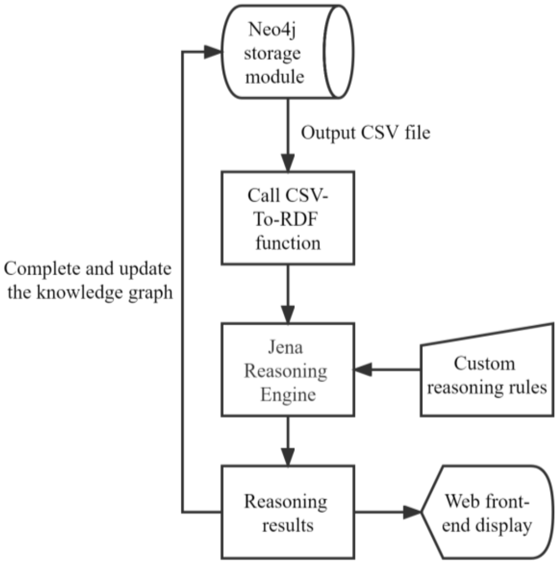


Figure 2. Execution reasoning process.

3. CONSTRUCTION OF KNOWLEDGE GRAPH

To establish a knowledge graph model in the workshop scenario, it is necessary to have a sufficient understanding of the production business process of the factory. We have carefully studied the production process of vehicle parts, which can be divided into the following steps: (a) Materials are transported to the automatic assembly workshop; (b) The assembly line in the assembly workshop starts production according to the product template; (c) The robotic arm performs assembly tasks; (d) The product is transported to the spraying workshop by the AGV(Automated Guided Vehicle) trolley; (e) The unloading robot starts spraying according to a fixed rhythm; (f) The product is transferred to the quality inspection workshop for testing. After getting familiar with the production process, we model the knowledge graph for the product quality problems in this scenario.

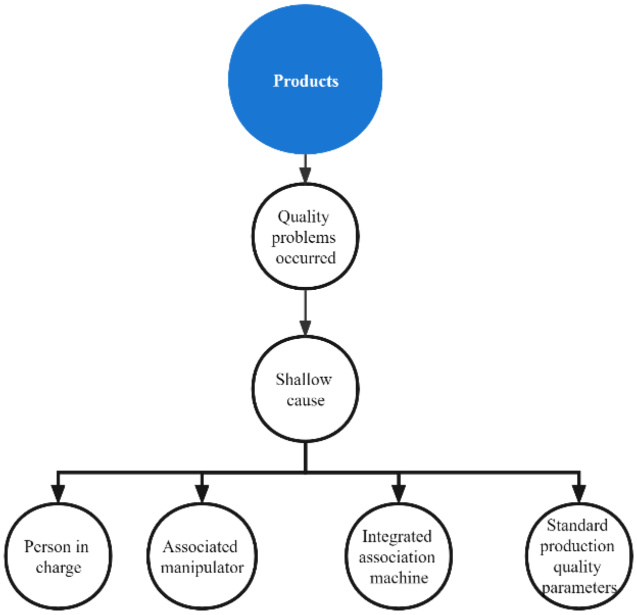

Products
Quality problems occurred
Shallow cause
Person in charge
Associated manipulator
Integrated association machine
Standard production quality parameters

Figure 3. Knowledge graph model.

The source of the metadata is the data export of the existing MES in the workshop, and the data is filtered through the Java program to obtain valuable information for the reasoning system. The final established knowledge graph model is shown in Figure 3. The knowledge graph is a top-down four-layer reasoning model. From the product with the problem to the direct cause at the shallow level and the indirect factor at the deeper level, the reason for the quality problem can be reasoned from the shallower to the deeper. The fine-grained can be specific. The production parameter information of a robot arm on a production line.

Furthermore, in this article, Neo4j is used as the graph database for storing knowledge graphs. Compared with the traditional relational database, its query speed is faster, the operation is simpler, and it can provide a richer presentation form. At the same time, Neo4j supports a complete transaction, that is, it satisfies the properties of ACID (Atomicity, Consistency, Isolation, Durability)[5]. In this system, the use of Neo4j will be reflected in the Java Spring boot microservice, and the graph database is completed by calling the relevant API (Application Programming Interface).

4. KNOWLEDGE REASONING METHOD

In the reasoning related to knowledge graph, there are mainly the following reasoning methods[6]:

(a) Description logic based reasoning (DL-based);

(b) Reasoning based on graph structure and statistical rule mining (PRA, AMIE);

(c) Reasoning based on knowledge graph representation learning (TransE);

(d) Probabilistic logic-based method (statistical relational learning);

(e) Production rule-based reasoning.

In our system, the production rule reasoning based on Jena reasoning engine is mainly used. The RDF (Resource Description Framework) model is used, which is the simplest and closest data model to natural language and human brain cognition[7]. The reasoning method based on production rules is mainly composed of fact set, rule set and reasoning engine.

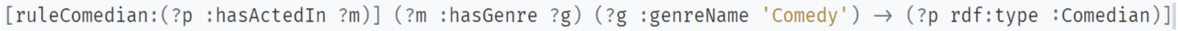
Apache Jena (Jena) is an API for Java Semantic Web Applications. In this paper, we use the inference engine based on the RETE algorithm in the Jena inference engine to carry out custom rule inference on the CSV files generated by the knowledge graph constructed in the previous article, and use the isolated entity nodes in the knowledge graph to generate new Relational attributes are connected to enrich and complement the knowledge graph.

After establishing the basic knowledge map model, we need to convert the knowledge graph into RDF file for reasoning operation by the reasoning engine. Because neo4j we can't directly generate RDF file, we need to divide the operation of generating RDF file into the following two processes, the first step is Neo4j-to-CSV file, and the second is CSV-to-RDF. In the Neo4j graph database, model files in CSV format can be exported, but direct dumping to RDF files[8] supported by the reasoning engine is not supported, so we need to convert the exported CSV files into RDF files. The CSV file export can be done using the plugin APOC (Awesome Procedures on Cypher) provided by Neo4j. At the same time, we need to install the desktop version of Neo4j locally and the APOC jar package. After the installation is successful, enter return apoc.verison() on the Neo4j browser operation page to check whether the installation is successful.

In addition, we need to add "apoc.export.file.enabled = true" to the Neo4j configuration file neo4j.conf before you can export the CSV file. Enter CALL "apoc.export.csv.all('yourcsv.csv', {})" in the browser operation interface to export all node and relational models in the current database and save them in the import folder under the installation directory. Jena provides related functions to convert CSV files, just call the CSV2RDF.init() function in the Java program, and we can create a new reasoning engine in Jena through RDFDataMgr.loadModel(".csv"). The reasoning model of Reference[9].

After creating a generative reasoning model in Jena, the model is traversed through the model.listStatements function to obtain all triplet information. Jena reasoning engine supports reasoning on RDF graph, providing forward chain, backward chain and mixed execution mode of both[10]. Contains a RETE (A kind of fast forward rule matching algorithm) engine[11] and one tabled datalog engine. Configuration parameters can be configured through "GenericRuleReasoner", and various reasoning engines can be used. To use a "GenericRuleReasoner", a ruleset is required to define its behaviour.

As shown in Figure 4, this is the specific format of a Jena rule, in which, assuming that in a movie knowledge graph, if the type of movie an actor participates in is a comedy, we can consider the actor to be a comedian[12].



```
[ruleComedian:(?p :hasActedIn ?m)] (?m :hasGenre ?g) (?g :genreName 'Comedy') → (?p rdf:type :Comedian)]
```

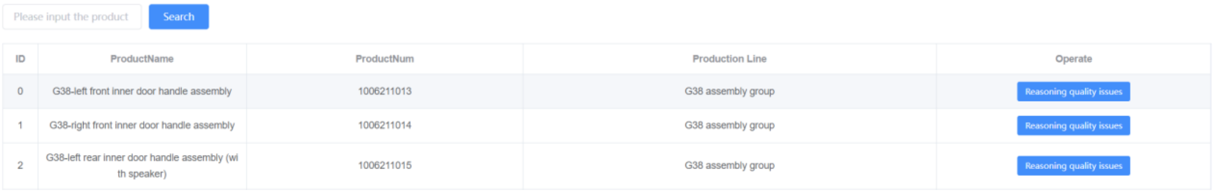
Figure 4. Jena's reasoning rules.

After defining the reasoning rules, we can call the List rules = Rule.rulesFromURL(XXX) method in the program to load and parse the rule file. In our paper, the process by which Jena performs reasoning is as follows: (a) Load RDF file using Input function; (b) Create a reasoning engine object via GenericRuleReasoner; (c) Parse reasoning rules using Rule.parseRules; (d) setMode sets the reasoning mode, this system adopts mixed reasoning; (e) Call the find function to perform reasoning and output; (f) Complete updating for Neo4j database.

5. SYSTEM DEMONSTRATION & PERFORMANCE

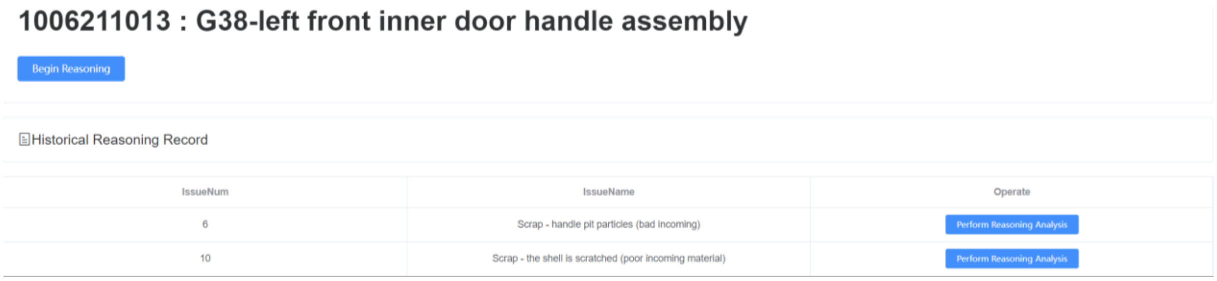
After completing the construction of the knowledge graph and the development of the inference engine in the previous article, we developed a set of web applications based on the Spring Cloud microservice framework, and transformed the work in the previous article into a practical application. A human-machine-friendly interactive page is developed based on the Vue framework for the factory production manager, providing the function of tracing the cause of quality problems and supporting follow-up feedback. Through the interactive page at the front end, we can vividly show this process. Next, the system is divided into several functional parts for testing.

Figure 5 lists all the products in production in the current factory and provides a search interface. Click the "Reasoning quality issues" button on the right to enter the next operation interface.



ID	ProductName	ProductNum	Production Line	Operate
0	G38-left front inner door handle assembly	1006211013	G38 assembly group	Reasoning quality issues
1	G38-right front inner door handle assembly	1006211014	G38 assembly group	Reasoning quality issues
2	G38-left rear inner door handle assembly (wi th speaker)	1006211015	G38 assembly group	Reasoning quality issues

Figure 5. Product list.


1006211013 : G38-left front inner door handle assembly

Historical Reasoning Record

IssueNum	IssueName	Operate
6	Scrap - handle pit particles (bad incoming)	Perform Reasoning Analysis
10	Scrap - the shell is scratched (poor incoming material)	Perform Reasoning Analysis

Figure 6. List of product issues.

Figure 6 shows the historical problems that have occurred in the currently selected product. Click perform reasoning Analysis on the right side of the problem to execute reasoning analysis of the cause of the problem. At the same time, click begins reasoning to select the quality issues found by the user and make a new one's reasoning analysis.

In Figure 7, according to the actual situation, we can select the quality problem that occurs during the quality inspection, and after clicking the confirm button, the reasoning analysis of this aspect will be performed.

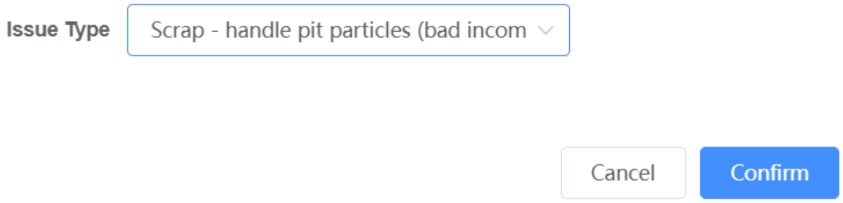
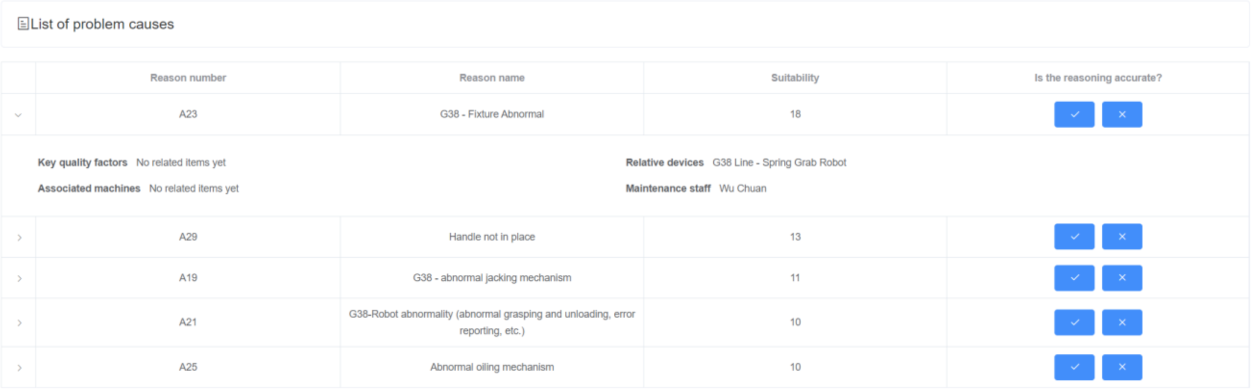

Issue Type: Scrap - handle pit particles (bad incom

Figure 7. Selecting quality issues.

After selecting the question type, the Jena reasoning analysis process can be executed, and the obtained results will be sorted by possibility and presented to the user for further investigation, as shown in Figure 8.


List of problem causes

	Reason number	Reason name	Suitability	Is the reasoning accurate?
⌄	A23	G38 - Fixture Abnormal	18	✓ ✕
	Key quality factors No related items yet Associated machines No related items yet		Relative devices G38 Line - Spring Grab Robot Maintenance staff Wu Chuan	
›	A29	Handle not in place	13	✓ ✕
›	A19	G38 - abnormal jacking mechanism	11	✓ ✕
›	A21	G38-Robot abnormality (abnormal grasping and unloading, error reporting, etc.)	10	✓ ✕
›	A25	Abnormal oiling mechanism	10	✓ ✕

Figure 8. Reasoning result presentation.

In the drop-down box of shallow direct causes, there are several deeper factors. Through the results given by the reasoning engine, the user can carry out secondary feedback confirmation, and the reasoning system will modify and supplement the knowledge map according to the feedback results, to make the next reasoning process more accurate.

6. CONCLUSION

Based on the Jena reasoning engine, this system implements an analysis system based on rule-based reasoning, and uses Neo4j as the storage medium of the knowledge graph to feedback and iterate the reasoning results to the graph database, forming a closed-loop system. The accuracy of the reasoning results has contributed to the production efficiency of the auto parts production line, reducing the time-consuming and laborious cost of manual inspection. In the current context of Industry 4.0, the application of knowledge graphs in the field of industrial production will bring a huge impetus to intelligent manufacturing.

In the next few years, knowledge graph will still be a frontier research hotspot in the field of big data intelligence. How to apply knowledge graphs to more disciplines is an important issue that we need to think deeply about. In the context of the Internet of Everything, no entity exists independently. All things in the objective world have a relationship with the outside world. How to abstract them and establish a more reliable model is also the focus of future research.

REFERENCES

[1] Yang, S., Wang, X., Zhao, Z., Pan, J. and Wu, L., “Research on construction and application of COVID-19 knowledge graph,” Journal of Qingdao University (Engineering Technology Edition), 36(4), 22-29(2021). (in Chinese)

[2] Chen, H., Zhang, W., Wong, C., Ye, G. Q., Wen, B. and Zhang, W., “Large scale pre-trained knowledge graph model and e-commerce application,” Big Data Research, 7(3), 97-115(2021). (in Chinese)

[3] Dong, H., “Research on the application of knowledge graph technology in bank network security,” Electronic World, (18), 63-64(2021). (in Chinese)

[4] Buchgeher, G., Gabauer, D., Martinez-Gil, J. and Ehrlinger, L., “Knowledge graphs in manufacturing and production: A systematic literature review,” IEEE Access, 9, 55537-55554(2021).

[5] Ballas, I., Tsakanikas, V., Pefanis, E. and Tampakas, V., “Assessing the computational limits of GraphDBs’ engines—A comparison study between Neo4j and Apache Spark,” 24th Pan-Hellenic Conference on Informatics, (2020).

[6] Wang, M., [Construction and Application of Academic Knowledge Graph], Xidian University, Xi’an, Master’s Thesis, (2021). (in Chinese)

[7] Gu, H. and Yu, Y., “Semantic internet application based on domain ontology and knowledge reasoning,” Journal of Shanghai Jiaotong University, (4), 583-585(2004). (in Chinese)

[8] Zhang, J., Shi, X. and Li, X., “General data access method based on ontology,” Electronic Technology and Software Engineering, (22), 216(2013). (in Chinese)

[9] Feng, H., Duan, L. and Zhang, B., “Review of knowledge reasoning oriented to knowledge graph,” Computer System Applications, 30(10), 21-30(2021). (in Chinese)

[10] Ma, M. and Chen, C., “Research on traffic ontology reasoning mechanism based on Jena development kit,” Henan Science and Technology, (13), 102-104(2020). (in Chinese)

[11] Shafi, U. F., Sattar, H., Bajwa, I. S. and Ikram, A., “Parsing RDFs to extract object oriented model using Apache Jena,” INTAP, 119-129(2018).

[12] Li, J., Yang, B., Ma, N., Liu, Y., Wang, D. and Chen, B., “Reverse reasoning technology of product key design quality characteristics based on fault feedback,” China Mechanical Engineering, 33(02), 176-186(2022). (in Chinese)