2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 6-10, 2024. Sarawak, Malaysia

Ontology Mapping-based Semantic Reasoning with OPC UA for Heterogeneous Industrial Devices

Jing Bi[1], Rina Wu[1], Haitao Yuan[2], Ziqi Wang[1], Jia Zhang[3] and MengChu Zhou[4]

[1]College of Computer, Beijing University of Technology, Beijing 100124, China
[2]School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[3]Dept. of Computer Science in Lyle School of Engineering, Southern Methodist University, Dallas, TX, USA
[4]Dept. of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA

2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 978-1-6654-1020-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/SMC54092.2024.10831352

***Abstract*—The advent of smart manufacturing in Industry 4.0 signifies the arrival of the era of connections. As an excellent communication protocol, Object linking and embedding for Process Control Unified Architecture (OPC UA) can address most semantic heterogeneity issues. However, its semantics are not formally defined at the application layer. To address the information silo problem caused by semantic heterogeneity, a method named Querying of Ontology Mapping-based OPC UA (QOMOU) is proposed. It extracts the information models of OPC UA servers into resource description framework triples, utilizes web ontology language for semantic enrichment and inference, and employs a semantic similarity model for event ontology mapping to improve query efficiency. The method's effectiveness is validated through functional queries using the SPARQL protocol in Apache Jena. The query efficiency is 5% higher on average compared to both structured query and extensible markup languages. Moreover, by employing a keyword-matching algorithm, the query accuracy of the existing heterogeneous data integration scheme is improved by 4% on average. This enhancement can boost the operational efficiency of Internet of Things systems based on the OPC UA architecture.**

***Index Terms*—OPC UA, ontology, syntactic interoperability, semantic similarity, semantic heterogeneity.**

I. INTRODUCTION

With the rise of Industry 4.0, communication between devices has become increasingly important. To achieve the Internet of Things, Object linking and embedding for Process Control Unified Architecture (OPC UA) emerges as a unified architecture for communication on the Industry 4.0 open platform [1]. OPC UA is a unified communication protocol that can facilitate communication between devices and systems from different vendors without caring about details of the underlying implementation. Fig. 1 illustrates the overview of vertical and horizontal communication. OPC UA standardizes communication from the field level to a unified information level, adding metadata to each data object to describe its type, structure, and characteristics [2].

However, interoperability issues may still arise between OPC UA products from different vendors, presenting challenges when integrating devices from various manufacturers. In other words, OPC UA meets the requirements of syntactic interoperability at the information layer. However, semantic interoperability at the information layer remains undetermined. Distributed data management and interoperability rely on specified ontologies between two or more machines at this point. These ontologies can automatically and accurately interpret the meaning of exchanged data and apply it to valuable objectives. For semantic interoperability, ontologies must consider the metadata exchanged between different systems and environments. Raising the level of semantic interoperability can better achieve communication between the control level and the enterprise level. In these communications, providing simple and convenient human-machine interaction interfaces enables even non-technical managers to clearly understand the operational status of the factories or processes through the interactive interfaces. This helps improve production efficiency [3], reduce costs [4], and enhance management decision-making.

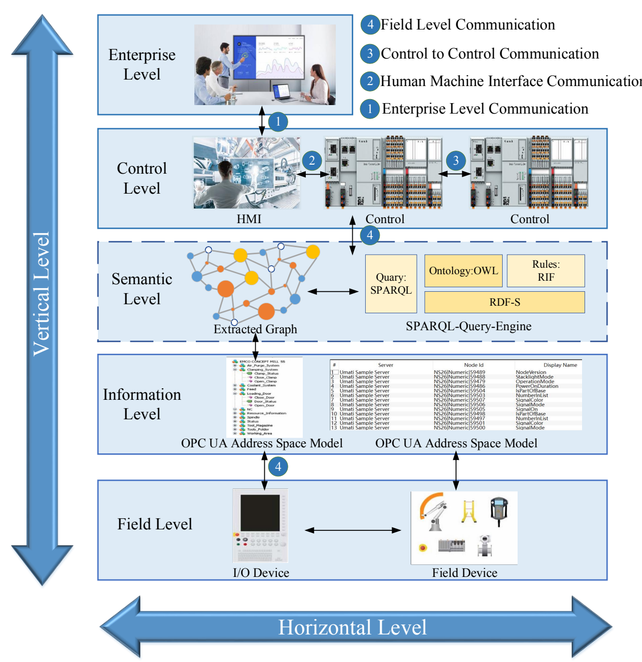
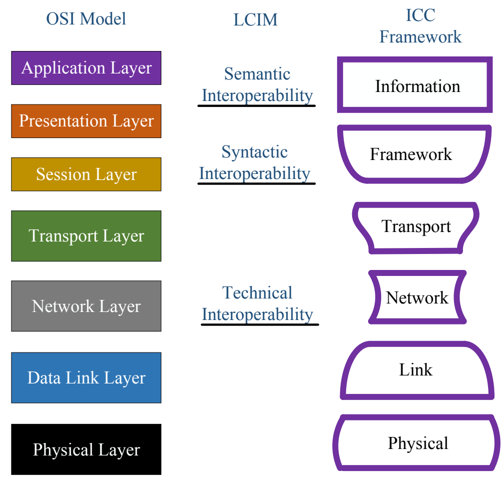


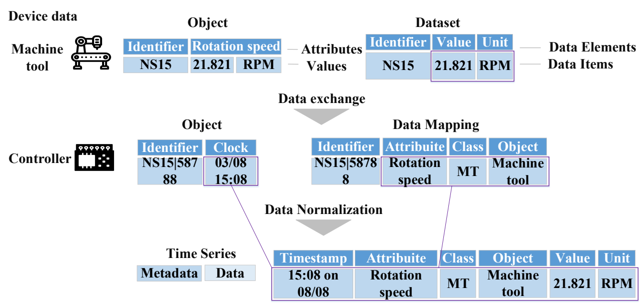
Fig. 1. Overview of vertical and horizontal communications.

The most significant obstacle faced by widespread applications of the Internet of Things and automation systems

This work was supported by the Beijing Natural Science Foundation under Grants L233005 and 4232049, and the National Natural Science Foundation of China under Grants 62173013 and 62073005.



(a) OSI models redefined by IICF.



(b) Factory data storage formats and standardization.

Fig. 2. Transition from syntax to semantic interoperability.

is semantic interoperability. Industry organizations are trying to implement semantic data models that cover a wide range of industries and systems [5], *e.g.*, the Object Management Group, Global Standards 1, and Schema.org. However, they are based on a series of industry segments, and they are not formally defined at the semantic level. In this case, the solution to semantic interoperability is to increase the meaning of data. Data from smart devices is stored and transmitted in multiple formats, with inconsistent and non-standardized naming conventions and limited descriptions to understand their meanings. Current studies aim to enrich the meaning of data to improve semantic interoperability. For example, Westermann *et al.* [6] propose a semantic model that formalizes variables in events and utilizes an RDF query language (SPARQL) for querying and accessing data. However, this approach does not provide a clear solution for dealing with heterogeneous data, and it does not validate the accuracy of the query. Bakakeu *et al.* [7] convert the OPC UA information model into web ontology language description logic, enabling automated reasoning in that language. However, they do not address the issue of handling heterogeneous information. In summary, current methods fail to address the issue of semantic heterogeneity well.

Based on the aforementioned analysis, this work proposes an ontology mapping approach named Querying of Ontology Mapping-based OPC UA (QOMOU) to tackle this fundamental problem. First, the OPC UA information model is extracted and converted into a graph structure. Then, the meaning of the data is supplemented using Web Ontology Language (OWL), providing clear structures and significant interpretation of data object meanings. Furthermore, this method determines whether data objects are the same by calculating the similarity between concepts explained by the data objects. Therefore, the heterogeneity [8] between data from different vendors is resolved by assessing the similarity of concept interpretations to determine if they represent the same object. Finally, the method's data management and reasoning capabilities are validated by inputting specific SPARQL queries into the Apache Jena engine's automatic reasoning mechanism [9] to obtain the desired answers. Comparative experiments on query efficiency and accuracy are conducted using different semantic encapsulation techniques and heterogeneous datasets. The results demonstrate the efficiency and effectiveness of the QOMOU.

II. BACKGROUND AND APPROACH

In this section, the primary issue addressed by Industry 4.0 is first discussed in II-A. Then, the implementation details of the QOMOU are presented in subsections II-B and II-C.

A. Semantic Interoperability and OPC UA

To achieve semantic interoperability across industries and domains, the Industrial Internet Consortium (IIC) redefines the traditional Open System Interconnection (OSI) reference model [10]. As shown in Fig. 2(a), combining the presentation layer and the session layer facilitates the structured parsing of data from endpoints. These improvements all follow the concept of the smart factory proposed by Industry 4.0. Although OPC UA has an information model, it does not fully utilize data for automated equipment management. Therefore, the meaning of the data needs to be understood. For example, in Fig. 2(b), when a machine tool obtains a value of 21.821, it is unclear whether it represents temperature, pressure, or other data types. If it represents a speed, it is also unclear whether it is rotational or operational speed, and the unit of this value is undetermined. Therefore, it requires further interpretation through data semantics [11].

B. Event Class Similarity Model

This subsection describes how to calculate the similarity of concepts between ontologies and how to avoid human bias. It details the extraction of similarity distribution tables for both event class elements [12] and structural similarity.

Event class is an abstract concept that can be defined as a six-tuple consisting of an action a, the participant or object o, the time of an event t, the location of an event p, and the state of an event s. It also can represent a group of events with common characteristics, *e.g.*, $C_1 = \{e_{11}, e_{12}, \cdots, e_{1n}\}$ and $C_2 = \{e_{21}, e_{22}, \cdots, e_{2m}\}$, where n and m denote the number of elements of C_1 and C_2, respectively. A classic set

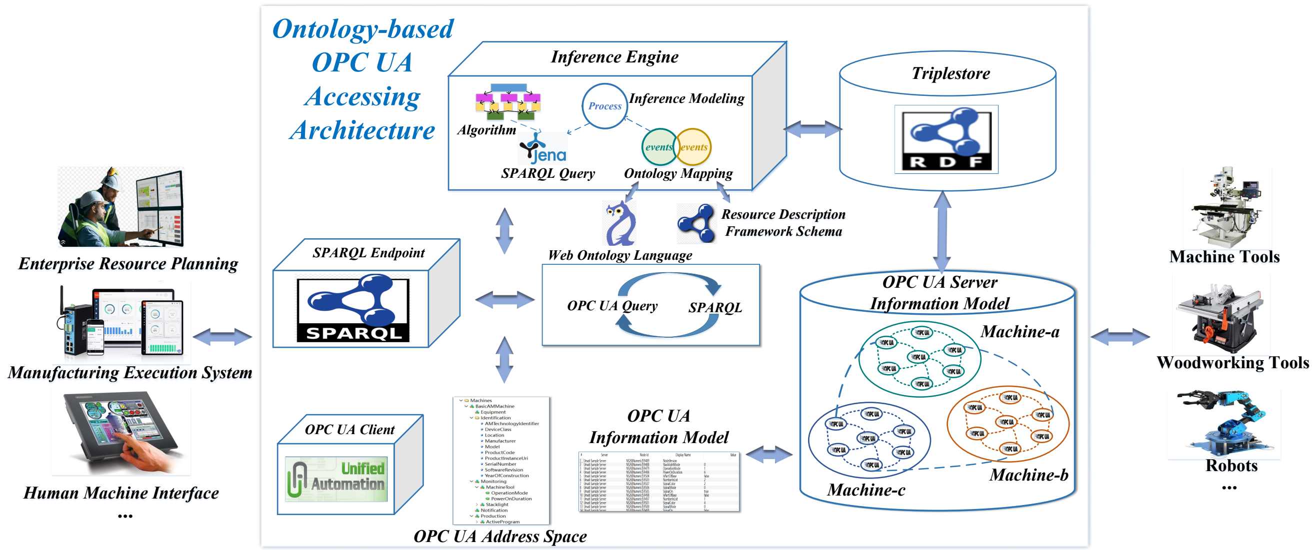


Fig. 3. Conceptual model validation of the integration of OPC UA with ontology, integrating semantic web technologies with OPC UA for sensor discovery and implementing a semantic access layer to harness the potential of ontologies without disrupting existing OPC UA standards.

similarity algorithm is proposed to calculate ontology similarity. The similarity of C_1 and C_2 is denoted as $s_o(C_1, C_2)$, *i.e.*,

$$s_o(C_1, C_2)=\frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} s_e(C_{1i}, C_{2j}) \tag{1}$$

where C_{1i} denotes element i in C_1 and C_{2j} denotes element j in C_2. $s_e(C_1, C_2)$ denotes the element similarity of C_1 and C_2. It is calculated by multiplying the syntactic and semantic similarities with their respective weights and then adding them together. *i.e.*,

$$s_e(C_1, C_2)=\sigma_q{\cdot}s_q(C_1, C_2)+\sigma_t{\cdot}s_t(C_1, C_2) \tag{2}$$

where σ_q and σ_t are the weights of syntax and semantic similarity, and $\sigma_q+\sigma_t=1$. Moreover, $s_q(C_1, C_2)$ and $s_t(C_1, C_2)$ denote the syntactic and semantic similarity between C_1 and C_2, and they are obtained from (3) and (4), respectively.

$$s_q\left(C_1, C_2\right)=\frac{2\sum_{i=1}^{n}\sum_{j=1}^{m}\phi\left(C_{1i}, C_{2j}\right)}{l\left(C_1\right)+l\left(C_2\right)} \tag{3}$$

where $\phi(C_{1i}$ and $C_{2j})$ is the longest common substring between two elements C_{1i} and C_{2j}, $l(\cdot)$ denotes the length of an event. In calculating semantic similarity, it is important to understand the concept of sememes. They are the smallest semantic units in language, serving as the basic elements that compose vocabulary and linguistic meaning. The formula for calculating the similarity of semantics is given as follows:

$$s_t(C_1, C_2)=\frac{|E_{c_1\longleftrightarrow c_2}|}{|c_1|+|c_2|}\cdot\left(\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}s_q(C_{1i}, C_{2j})}{|P_{c_1\longleftrightarrow c_2}|}\right) \tag{4}$$

where c_1 and c_2 are two sets of sememes, $|c_1|$ and $|c_2|$ are the numbers of sememes in the sets, $|E_{c_1\longleftrightarrow c_2}|$ is the number of sememes with semantic relationships in two sets, $|P_{c_1\longleftrightarrow c_2}|$ is the number of pairs from semantic sets with semantic relationships. Finally, the semantic similarity between C_1 and C_2 is denoted as $s\left(C_1, C_2\right)$, which is given as:

$$s\left(C_1, C_2\right)=\sum_{k=1}^{4}\eta_k\prod_{d\in[o,e,q,t]}s_d\left(C_1, C_2\right) \tag{5}$$

where η_k denotes the degree of impact of each part on the entire system.

The final similarity is calculated by multiplying the values of (1)–(4) by their respective weights and then summing them together. To avoid biases caused by human factors, QOMOU adopts a sigmoid function ($\sigma(x)$) to calculate the weights, *i.e.*,

$$\sigma(x)=\frac{1}{1+e^{-5(x-\alpha)}} \tag{6}$$

where x is the value of syntactic similarity or semantic similarity, -5 is a constant that controls the smoothness of the curve, which helps to avoid the generation of outliers, and α is a parameter that controls the symmetric center position of the curve.

C. Process of the QOMOU

In the previous subsection, the calculation of concept similarity is discussed and this subsection discusses the integrating process of the OPC UA information model with the ontology theory. As shown in Fig. 3, the OPC UA server [13] utilizes a simulated OPC UA sample server from UMATI[1], which includes information models for machine tools, woodworking tools, robotic arms, *etc*. OPC UA clients can be utilized for simple interactions. Moreover, the OPC UA information model can be extracted through an automated node crawler named Lion2[2]. After being converted to OWL,

[1]https://doi.org/10.5281/zenodo.6336935
[2]https://github.com/hsu-aut/lion

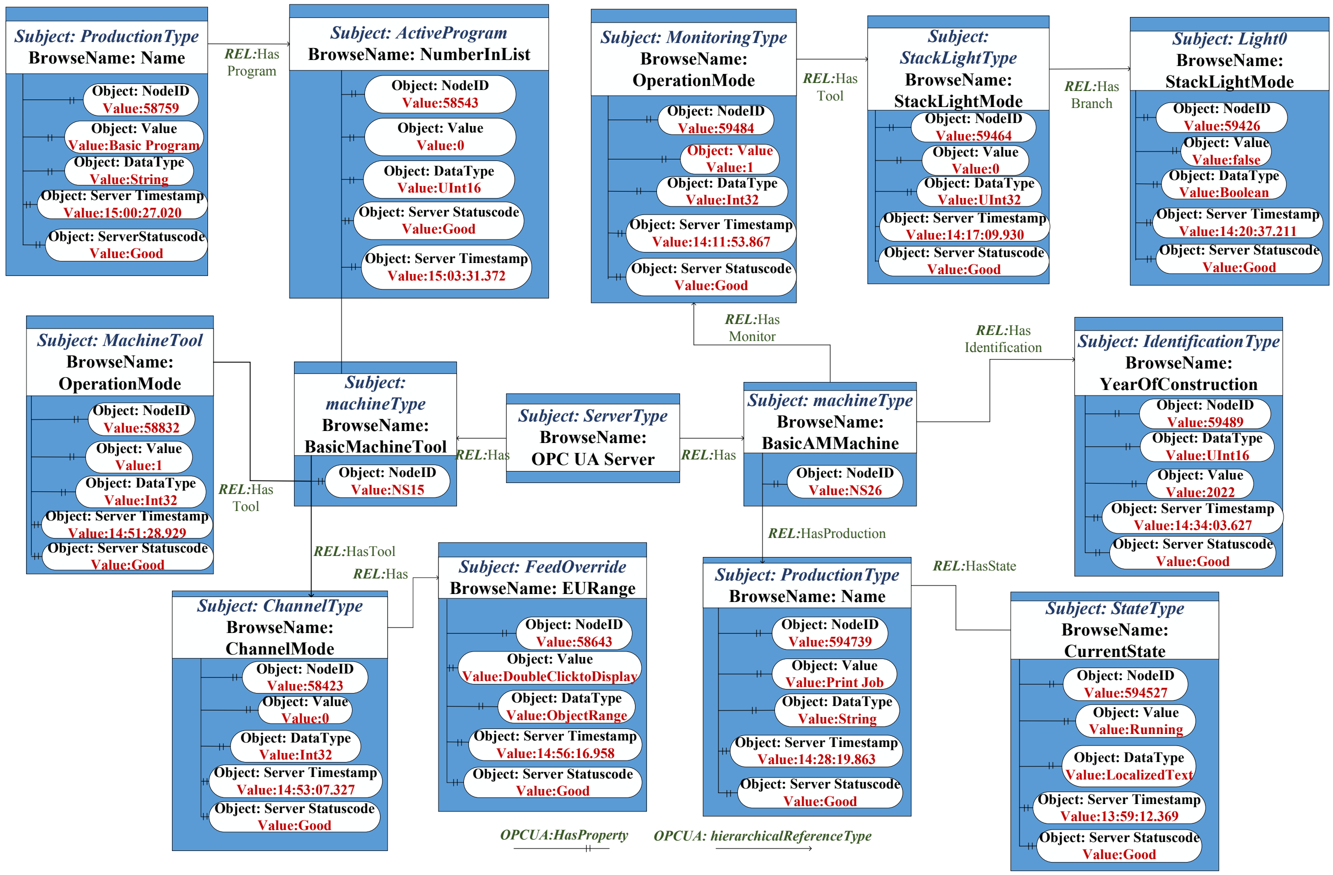


Fig. 4. Example instance nodes of OPC UA servers.

the similarity calculation method for event classes mentioned in the II-B is used to merge and classify various data objects. It can be adopted to convert it into OWL [14]. As shown in Fig. 4, various categories of events are classified from the server, each extracted as a subclass. These extracted events are then stored in a Triplestore[3] and fed into the inference engine. Then, SPARQL queries are used instead of OPC UA queries, enabling various interactions such as human-machine interface.

III. Experimental Evaluation

To validate the effectiveness of QOMOU, the capability of this model is first validated, including the relationship between devices and their components, the ability to monitor device event abnormalities, and the ability to handle such abnormalities.

A. Model Capability Validation

To answer the question "Which variables belong to a certain machine?", in an OWL model without semantic reasoning, a significant issue arises where the components in the third level in Fig. 4 cannot be categorized into the first-level equipment types. This means that the relationships between superclasses and subclasses [15] must be manually annotated, resulting in increased data volume and difficulty in querying. Therefore, a class hierarchy inference and reverse reasoning are introduced to ensure query completeness. The query statement is shown in Listing 1, and the OWL reasoning is shown in Listing 2. The query results of the part ownership relationship are shown in Fig. 5. The results demonstrate that by using the OWL inference engine, subclasses within OPC UA nodes can be automatically recognized as superclasses without the need to specify the relationships between them explicitly.

```
SELECT ?variable ?nodeID ?variableName ?variableType ?machineID
?machineName
#Querying information from OWL
WHERE {
?urnid owl:sameVariableAs OpcSS:AllMachines.
#Selecting variables from OPC UA machines.
?machine OpcUa:hasComponent∗ ?variable.
?machine OpcUa:browseName∗ ?machineName.
?variable OpcUa:belongTo ?machineID.
?variable OpcUa:browseName ?variableName.
?variable OpcUa:typeDefinition ?variableType.
#Filtering variable types.
FILTER( ?variableType = "IdentificationType" ||?variableType =
      "ProductionType" ||?variableType = "MonitoringType").
#Querying from a Specific Time Point
?node OpcUa:histValues ( ?Time ?Value "2024–03–16T08:00:00Z"
      "2024–03–28T08:20:00Z").
}
```

[3]https://jena.apache.org/

```
LIMIT 5
```

Listing 1. Query statement-part A.

```
@prefix :<http://www.zenodo.org/records/.com#> .
@prefix owl: <http://www.zenodo.org/2002/07/owl#> .
@prefix rdf: <http://www.zenodo.org/1999/02/22−rdf−syntax−ns#> .
@prefix xsd: <XML Schema> .
@prefix rdfs: <http://www.zenodo.org/2000/01/rdf−schema#> .
#Subclass and Superclass Reasoning
[ruleBelongtoMonitoringType: (?p :hasMonitor ?m) (?m :hasTool ?g) (?g
    :hasBranch ?q) (?p browseName: 'BasicAMMachine')−> (?q
    rdf:belongto :p)]
#Inverse Reasoning of Inclusion Relationship
[ruleInverse: (?p :hasVariable ?m) −> (?m :belongTo ?p)]
```

Listing 2. OWL reasoning.

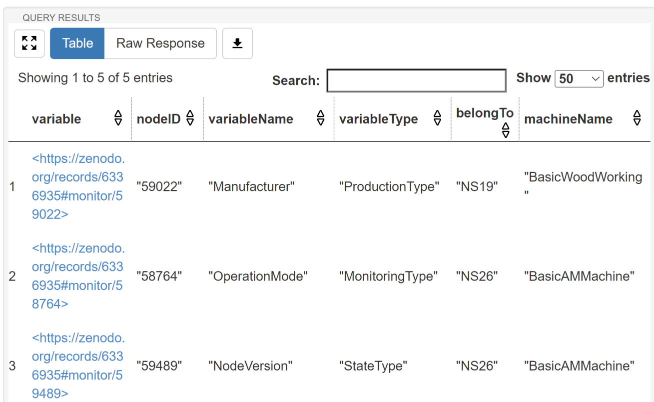

QUERY RESULTS

Table | Raw Response

Showing 1 to 5 of 5 entries Search: Show 50 entries

	variable	nodeID	variableName	variableType	belongTo	machineName
1	<https://zenodo.org/records/6336935#monitor/59022>	"59022"	"Manufacturer"	"ProductionType"	"NS19"	"BasicWoodWorking"
2	<https://zenodo.org/records/6336935#monitor/58764>	"58764"	"OperationMode"	"MonitoringType"	"NS26"	"BasicAMMachine"
3	<https://zenodo.org/records/6336935#monitor/59489>	"59489"	"NodeVersion"	"StateType"	"NS26"	"BasicAMMachine"

Fig. 5. Query results of the part ownership relationship.

When the device encounters an abnormal event, the architecture queries the issue and raises an abnormal alarm. The code for querying anomalies is displayed in Listing 3, and the detection results are shown in Fig. 6. The results demonstrate that the architecture can remind the staff to switch the mode from automatic to manual, which helps to prevent an abnormal situation during the production process.

```
SELECT ?event ?eventType ?value ?timestamp ?statuscode
?machineName
WHERE {
#Selecting events from BaiscAMMachine Procedure and Unit
?proc ISA88:isMonitoringInProcessStage ?Process .
?urnid owl:isAssigneTo ?Process .
FILTER( ?proc = OpcSS:UnitProcedureWarning).
FILTER( ?unit = OpcSS:BasicAMMachine).
#Selecting timestamps of events
?Process ISA88:hasInput ?starttime.
?stimeDE Alarm59031:hasDescription
OpcSS:StartTimeProcess;
Error59029:hasDescription /
CurrentMode:Value ?starttime.
#Selecting events from OPC UA machines.
?machine OpcUa:hasEvent∗ ?event .
?event OpcUa:type∗ ?eventType .
?event OpcUa:hasValue ?value .
?event OpcUa:monitorTime ?timestamp .
?event OpcUa:state ?statuscode .
FILTER(?variableType = "State"||?variableType = "MonitoringType") .
#Extracting from RDF information
?node OpcUa:histValues ( ?Time ?Value ?starttime ?endtime) .
}
```

Listing 3. Query statement-part B.

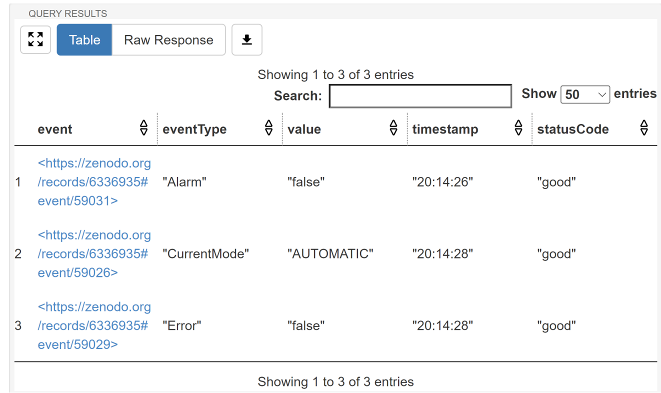

QUERY RESULTS

Table | Raw Response

Showing 1 to 3 of 3 entries Search: Show 50 entries

	event	eventType	value	timestamp	statusCode
1	<https://zenodo.org/records/6336935#event/59031>	"Alarm"	"false"	"20:14:26"	"good"
2	<https://zenodo.org/records/6336935#event/59026>	"CurrentMode"	"AUTOMATIC"	"20:14:28"	"good"
3	<https://zenodo.org/records/6336935#event/59029>	"Error"	"false"	"20:14:28"	"good"

Showing 1 to 3 of 3 entries

Fig. 6. Query results of the part anomalies.

B. Efficiency and accuracy of queries

In this subsection, the experiments on the query efficiency and query accuracy of QOMOU are conducted to validate its effectiveness [16]. To compare query efficiency, SQL datasets with and without format wrapping and XML-based datasets are selected. Moreover, 4,000 OPC UA server data are extracted, with query times recorded for every 300 pieces of data. Fig. 7 shows the comparison of query times for different semantic web technologies.

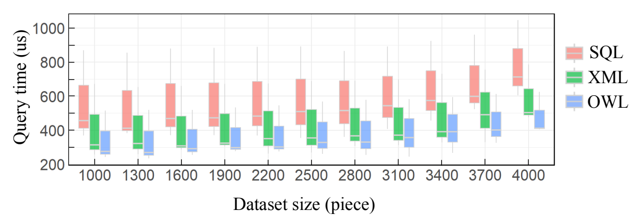


Fig. 7. Query time for different semantic web technologies.

It is shown that the query time of the QOMOU is smaller than that of SQL and XML semantic integration schemes, validating its effectiveness. In validating the query accuracy rate, the accuracy [17] of existing heterogeneous data integration schemes is compared. The accuracy metric is QAR, where the query result set is R, and the users' expected result set is S_R, S_R is calculated as:

$$QAR = \frac{S_R}{R} \times 100\% \tag{7}$$

Moreover, experiments are conducted to validate the query accuracy of QOMOU [18]. The comparison methods include probability factor framework [19] and logical object-oriented interaction [20]. Furthermore, the experimental data is divided into four groups, each with two hundred thousand data points. This ensures that accurate fluctuation comparisons can be made after completing the experiments for all four groups.

TABLE I
QUERY ACCURACY ANALYSIS

Methods	Group 1	Group 2	Group 3	Group 4	Float Ratio (%)
Probability factor framework	0.60	0.49	0.58	0.68	3.2∼16
Logical object-oriented interaction	0.69	0.62	0.57	0.56	6.5∼19
QOMOU	**0.71**	**0.65**	**0.63**	**0.60**	**2.1∼10**

Table I shows the accuracy and fluctuation ranges of three methods. It is illustrated that the accuracy of each method fluctuates within a certain range. Due to the utilization of event-class semantic integration algorithms, the accuracy of QOMOU is above average, indicating higher query accuracy.

IV. CONCLUSION

Industry 4.0 transforms traditional manufacturing into intelligent manufacturing. This work emphasizes the importance of semantic interoperability in factory workshops. Moreover, Object Linking and Embedding for Process Control Unified Architecture (OPC UA) is an effective knowledge model suitable for factory workshops. However, due to data heterogeneity, achieving unified management and communication in factories with OPC UA is extremely challenging. Currently, ontology mapping methods based on OPC UA suffer from slow speed and low efficiency. Therefore, a Querying of Ontology Mapping-based OPC UA (QOMOU) is designed to solve the above problems. QOMOU maps the OPC UA information model to the resource description framework in ontology technology and creates a web ontology language in a graph structure. Additionally, it calculates the similarity of event class concepts and addresses the semantic heterogeneity issue of devices produced by different manufacturers. Finally, the effectiveness of QOMOU is validated through functional queries. Experimental results show that the query time with QUMOU is reduced by 5% on average than those with SQL and XML. Furthermore, QUMOU achieves a 4% on average higher query accuracy than other state-of-the-art querying models.

In the future, we plan to refine our semantic mapping algorithm to enhance the accuracy of queries by integrating similarity models of event classes. We also plan to utilize dynamic database storage for mapping models to monitor OPC UA servers in real-time.

REFERENCES

[1] A. Busboom, "Automated Generation of OPC UA Information Models–A Review and Outlook," *Journal of Industrial Information Integration*, vol. 39, no 3, pp. 100602–100617, Mar. 2024.

[2] H. Abukwaik, A. Gogolev, C. GroB, and M. Aleksy, "Information Models for a New Generation of Manufacturing Systems–a Case Study of Automated Guided Vehicle," *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2023, Italy, pp. 372–278.

[3] J. Bi, Z. Wang, H. Yuan, J. Zhang, and M. Zhou, "Cost-Minimized Computation Offloading and User Association in Hybrid Cloud and Edge Computing," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2024.3354348.

[4] J. Bi, Z. Wang, H. Yuan, J. Qiao, J. Zhang, and M. Zhou, "Self-adaptive Teaching-learning-based Optimizer with Improved RBF and Sparse Autoencoder for Complex Optimization Problems," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, 2023, pp. 7966–7972.

[5] R. Xu, W. Mayer, H. Chu, Y Zhang, H. Zhang, Y. Wang, Y. Liu, and Z. Feng, "Automatic Semantic Modeling of Structured Data Sources with Cross-modal Retrieval," *Pattern Recognition Letters*, vol. 177, no.7, pp. 7–14, Jan. 2024.

[6] W. Tom, H. Nemanja, and F. Alexander, "Accessing and Interpreting OPC UA Event Traces based on Semantic Process Descriptions," *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, Germany, pp. 1–7.

[7] J. Bakakeu, M. Brossog, J. Zeitler, J. Franke, S. Tolksdorf, H. Klos, and J. Peschke, "Automated Reasoning and Knowledge Inference on OPC UA Information Models," *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, 2019, Taipei, Taiwan, pp. 53–60.

[8] B. Wang, B. Song, Y. Li, Q. Zhao, and B. Tan, "Towards Smart Sustainable Cities: Addressing Semantic Heterogeneity in Building Management Systems Using Discriminative Models," *Sustainable Cities and Society*, vol. 62, no. 6, pp. 102367–102387, Jul. 2020.

[9] A. Patel, V. Tiwari, M. Ojha, and O. Vyas, "Ontology-based Detection and Identification of Complex Event of Illegal Parking Using SPARQL and Description Logic Queries," *Data & Knowledge Engineering*, vol. 226, no. 5, pp. 120149–120167, Oct. 2023.

[10] X. Zhang, and X. Ming, "Implementation Path and Reference Framework for Industrial Internet Platform (IIP) in Product Service System Using Industrial Practice Investigation Method," *Advanced Engineering Informatics*, vol. 51, no 6, pp. 128–134, Oct. 2022.

[11] T. Krijnen, and J. Beetz, "an Efficient Binary Storage Format for IFC Building Models Using HDF5 Hierarchical Data Format," *Automation in Construction*, vol. 113, no. 4, pp. 103134–103146, Mar. 2020.

[12] A. Algergawy, R. Nayak, and G. Saake, "Element Similarity Measures in XML Schema Matching," *Information Sciences*, vol. 180, no. 24, pp. 4975–4998, Aug. 2022.

[13] O. Sande, M. Fojcik, R. Cupek, and M. Drewniak, "Implementation of OPC UA Communication Traffic Control for Analog Values in an Automation Device with Embedded OPC UA Servers," *Procedia Computer Science*, vol. 225, no. 4, pp. 2322–2332, Aug. 2023.

[14] B. Idrissi, S. Baïna, A. Mamouny, and M. Elmaallam, "RDF/OWL Storage and Management in Relational Database Management Systems: a Comparative Study," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 7604–7620, Oct. 2022.

[15] S. Wei, M. Luo, Z. Yang, and X. Lin, "Using Object-oriented Coupled Deep Learning Approach for Typical Object Inspection of Transmission Channel," *International Journal of Applied Earth Observation and Geoinformation*, vol. 347, no. 4, pp. 113864–11397, Mar. 2024.

[16] J. Bi, Z. Wang, H. Yuan, J. Zhang, and M. Zhou, "Cost-Minimized Partial Computation Offloading in Cloud-Assisted Mobile Edge Computing Systems," *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Honolulu, Oahu, HI, USA, 2023, pp. 5052–5057.

[17] V. Christou, K. Koritsoglou, G. Ntritsos, G. Tsoumanis, G. Tsipouras, N. Giannakeas, E. Glavas, and T. Tzallas, "Heterogeneous Hybrid Extreme Learning Machine for Temperature Sensor Accuracy Improvement," *Expert Systems with Applications*, vol. 203, no. 4, pp. 113864–11397, Sept. 2024.

[18] H. Li, and Y. Yang, "Keyword Targeting Optimization in Sponsored Search Advertising: Combining Selection and Matching," *Electronic Commerce Research and Applications*, vol. 56, no. 4, pp. 117488–117501, Oct. 2022.

[19] Y. Lina, S. Quan, N. Anne, and L. Xue, "Things of Interest Recommendation by Leveraging Heterogeneous Relations in the Internet of Things," *ACM Transactions on Internet Technology (TOIT)*, vol. 16, no. 4, pp. 1–25, Nov. 2016.

[20] X. Bin, T. Kanter, and R. Rahmani, "Logical Interactions for Heterogeneous Iot Entities Via Virtual World Mirrors in Support of Ambient Assisted Living," *Journal of Ambient Intelligence and Smart Environments*, vol. 8, no. 5, pp. 565–580, Oct. 2016.