

# Unified Structured ID and Orthogonal Information Retrieval Technique on Chatroom-style Real-time Messaging Platform for Industrial Internet of Things

Koji Amano <sup>\*</sup>, Hiroyasu Nishiyama <sup>\*</sup>, Minoru Koizumi <sup>\*</sup>,  
Kazunobu Fujii <sup>†</sup>, Naoya Sudo <sup>†</sup>, Shunichi Tano <sup>‡</sup>

## Abstract

The Industrial Internet of Things (IIoT) has rapidly become a popular concept, but this concept has caused problems in managing consistent identifiers. To solve these problems in control system management, we propose Structured IDs, which can define the complicated structure of control systems using IDs on a tree structure, and relational links, which can describe meaning for subsets of structured IDs. Furthermore, in use cases in which consistently managed information is utilized, a method of distributing appropriate information to each stakeholder is required. To address this requirement, we propose an orthogonal information retrieval technique for Structured IDs. Finally, we verify that the proposed techniques can be implemented using web technology and demonstrate that they can solve problems in an IIoT use case.

*Keywords:* Control system management, Identifiers, Industrial Internet of Things, Information Retrieval.

## 1 Introduction

The IIoT (Industrial Internet of Things) era has arrived. The IIoT is rapidly becoming one of the most popular concepts in factory automation, control system management, and advanced predictive maintenance. However, the IIoT concept is causing problems in the management of consistent identifiers. Additionally, various stakeholders want to utilize IIoT data as consistently managed data. To solve these problems, we set three requirements on which the identifier must be based by investigating the problems in accordance with dramatic changes in control system management. (1) Uniqueness, (2) Concreteness, and (3) Attachability. To develop the identifier in accordance with these requirements, we propose a technique named Structured IDs. In addition, to define the meaning among Structured IDs and/or networks that Structured ID cannot define, we propose another technique named relational links. Furthermore, in use cases in which consistently managed information is utilized, a method of distributing appropriate information to each stakeholder is required. To address this requirement from stakeholders who have each KPI

---

<sup>\*</sup> Hitachi, Ltd. Center for Technology Innovation, Kanagawa, Japan

<sup>†</sup> Hitachi Power Solutions Co., Ltd., Ibaraki, Japan

<sup>‡</sup> The University of Electro-Communications, Tokyo, Japan

(key performance indicator), we propose an orthogonal information retrieval technique for Structured IDs on a chatroom-style real-time messaging platform.

## 2 Problems due dramatic changes in control system management

In this chapter, we describe the problems due to dramatic changes in control system management. Control system management involves giving names to target facilities and computer systems and then monitoring and/or controlling those targets in order to manage them properly. The targets in control systems are composed of many disparate identifiers, such as facility, area (abstract identifiers), machine, network, sensors, and identifier of failure. Most management software manages and shows the temperature, vibration, rotation speed, and managed status of the control object using numbers or letters to manage the control systems. SCADA (supervisory control and data acquisition) is common software that plays an important role in managing the target in a control system. SCADA shows pictures of a monitoring target and indicates its values and operating conditions. In addition, it shows a color-coded display of normal or abnormal conditions. SCADA also sends instructions toward the managed target. However, the current technique in SCADA is increasingly unable to manage its targets because of the following four problems.

### 2.1 Identifier collision due to intensive information

The first problem is identifier collision due to the increasing multiprovider based individual ID and information integration. If an ID integration function is injective, then it maps every ID to the integrated ID set where the ID is stored and no special exceptional function is needed. However, this is often not possible. When two IDs map to the same ID in an ID set, the result is called a “collision.” Thus, if an integrated ID function is not injective, collisions occur. When collisions occur, we store the ID for one of the ID sets and must find some other locations for the other IDs. In current control management systems, the problem is solved by making incompatible SCADA independent from the independent management systems (see Figure 1).

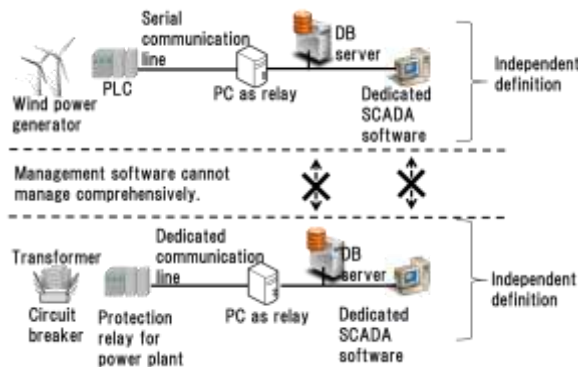


Figure 1: Independent SCADA for multiple architectures.

Current management software cannot manage individual information defined by each provider. Therefore, the operators need to aggregate that information in accordance with their knowledge and skills. Furthermore, in the IIoT concept, although information is aggregated to be utilized in IIoT, collisions occur in identifiers and various data cannot be stored in the data table.

## 2.2 Management of information uniquely tailored to stakeholders

The second problem is the increasing development cost of control system management software and the need to develop uniquely tailored information management systems due to the diversification of stakeholders. The SCADA designers need to extract, transform, and load information to fit it in the various levels of detail that diversified stakeholders want (see Figure 2).

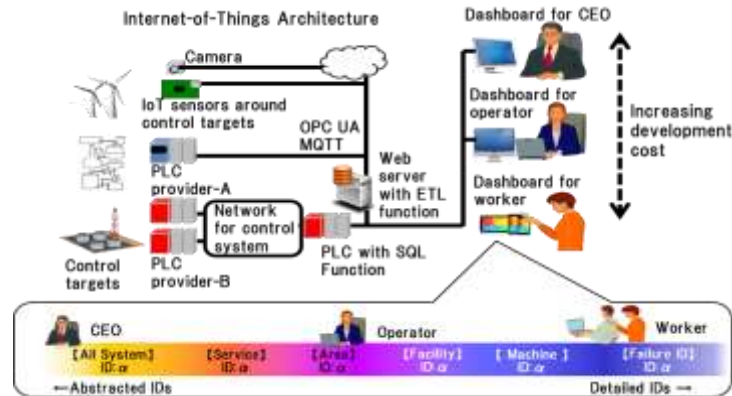


Figure 2: Various detail levels of information in IIoT platform.

In a web server (see Figure 2, left side and center), a management technique is required that can identify and consistently manage information. On the other hand, in the user side (see Figure 2, right side), filtering and information retrieval techniques are required that can fit various stakeholders' KPIs.

## 2.3 Example of synthesis and relationship management of definitions at different design phases

The third problem presents two challenges that structured IDs cannot solve.

### (1) Deficiencies in relations of meaning

Most multiprovider architecture lacks ways to define relationships in which different identifiers have the same health status and same failure. In addition, the architecture lacks ways to define the meaning of raw data that is generated by attachable IIoT sensors after the start of service.

### (2) Deficiencies in relations of networking

We need to define a “loop network structure” and/or “cross-coupled network structure.” Therefore, a flexible method is required for defining the relationships of networks.

## 2.4 Inflexible implementation method

As shown in Figure 1, sensing values are translated for each protocol by each electronic component, for example PLC (programmable logic controller), PC (personal computer) as relay, and DB (database) server. The data is passed through sequentially by each electronic component. The implementation methods have become inflexible due to wasteful conversion operations occurring redundantly.

### 3 Conventional methods

Control system management software uses several conventional methods.

The first method stores information in the DB by direct SQL (structured query language) by using an instrumentation device and a PLC that has high functionality.

Some papers proposed messaging techniques via a DB [1]-[4] between a PLC and views. However, their PLC is based on the assumption that all information of the control system is known by the controller. This means that expansion of the management area and integration of information are not assumed.

The second method uses an ETL function. ETL stands for extract, transform, and load, which are three database functions. To integrate information, various data on DBs are combined into one data format through pulling (extracting) data from the database, transforming it into a unified format, and then placing the format into another database. ETL can avoid collisions of identifiers and redefine the level of detail about target information. However, the control system management software must be reprogrammed to understand the new type of information.

The ETL function makes a new table and/or creates a virtual table to integrate information. This means that the ETL cannot manage various data in one type of data model. Thus, ETL functionality does not meet the real-time requirement for the control system management.

Various IoT system management designs have been researched [5]-[14] that use the NoSQL database to achieve flexibility. In studies of the data structure, research data management [15] with flexibility using XML [16] and XPath [17] has been proposed.

### 4 Our approach

After examining the problems and current techniques, we found that similar deficiencies in the systems provided by multiple providers include poor techniques to identify whole information that the operator wants to manage. There are no techniques to select that target information and then define meaning for the chosen subset. To solve this problem, we set three requirements for control system management on which the identifier must be based.

(1) Uniqueness

The information representing a feature of interest of the control system must be identified.

(2) Concreteness

The name of an abstract concept, equipment, parts, and failure ID must be managed on a unified platform, and detailed IDs must be defined gradually.

(3) Attachability

In addition to satisfying (1) and (2), partially defined identifiers must be joined together.

In this study, when we design an identifier that meets these requirements, we obtain hints from URLs (Uniform Resource Locators) on the Internet. A URL consists of the domain and site name that has a human readable string-based name. The structure of a URL is a joined domain name, folder name such as location, and file name. These lists of names gradually narrow down to the target IDs. On the basis of inspiration from URLs, we design a Structured ID as an identifier to meet the above three requirements.

Additionally, in a multiprovider system, since control system management software manages incompatible IDs and/or networks, we propose relational links as an additional concept to define meaning of relations among all IDs. These relational links will also be able to define meanings of subsets of Structured IDs.

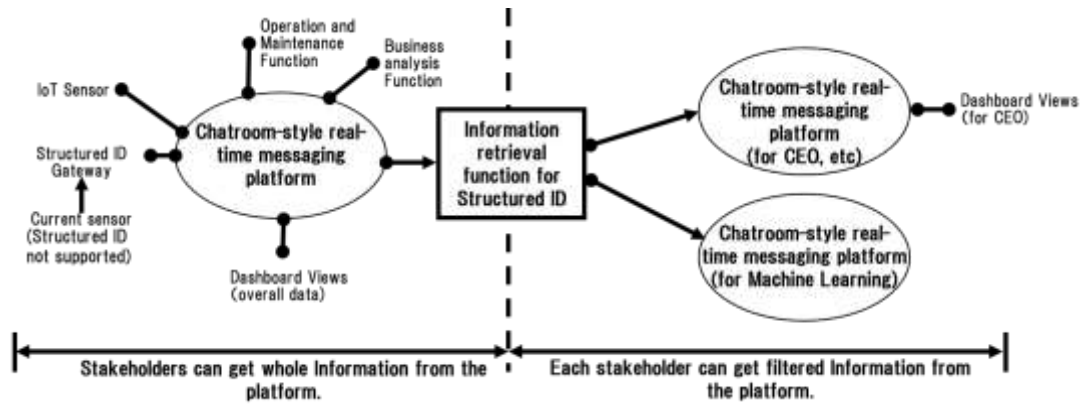


Figure 3: Implementation concept for real-time messaging platform and information retrieval function.

In implementation, for flexible control system management, we apply an information retrieval technique that is inspired by XPath. We expand from single dimensional retrieval among Structured IDs to multidimensional searching. Each stakeholder can share whole information and can receive specific information by a filter that has the proposed information retrieval technique on a chatroom-style real-time messaging platform (see Figure 3).

## 5 Managing IIoT identifier and its meaning

In this chapter, we first discuss the design of Structured IDs and then design relational links.

### 5.1 Design of Structured IDs to identify data

The details of system management for control systems were described in full in Chapter 2. The role of control system management software is to define names for target information and then manage its integrity. Our proposed technology is an ID design method to improve SCADA, and a Structured ID includes information for identifying the site of the system. Thus, we will use a different notation for Structured IDs to define them more easily. We denote a Structured ID by writing down its identifier of elements, separated by “/” slashes in Figure 4.

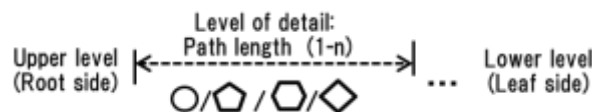


Figure 4: Definition model for Structured ID

The definition model makes detailed IDs gradually for Structured ID notation that has directivity. The following explains the approach for Structured IDs that meet the three requirements.

#### 5.1.1 Uniqueness

An ID is a unique string of characters. An ID also has sufficient information of a query to find the piece of target information with the highest precision and recall. The abstract model for Structured IDs in Figure 4 is a notation for enumerating paths in a collection of Structured IDs. A system designer needs to define Structured IDs to find the piece of target information on the basis of the path from the root ID to target ID.

### 5.1.2 Concreteness

Detailed Structured IDs are made by joining lower-side identifiers to the upper-side abstract Structured ID designed by a system integrator (SIer). For example, system designers can define IDs as follows in accordance with the concept in Figure 4: “Factory identifier. System identifier. Device identifier. Failure identifier.”

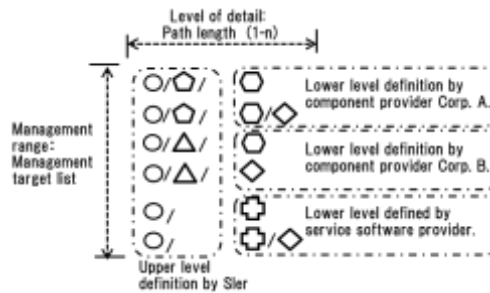


Figure5: Tree of Structured ID

When system designers make long Structured IDs, they might find that those Structured IDs consist of the same partial list of Structured IDs in the upper side and different partial lists of Structured IDs in the lower side. This means we can make a tree structure by collecting each Structured ID (see Figure 5).

### 5.1.3 Attachability

Design timings by the SIer, component providers, and machine providers include various design phases and definition of an identifier of failure. The proposed solution for this problem in the design phases is to create two ID subsets. The first subset is the upper-side subset that is defined by a SIer, and the second subset is the lower-side subset that is defined by a component provider. These two subsets of Structured IDs can be attached as shown in Figure 6.

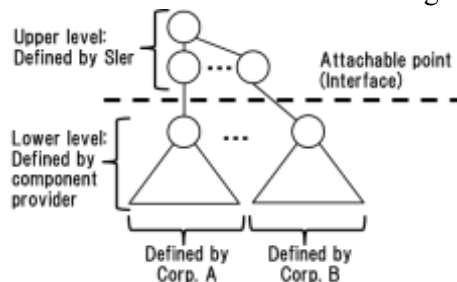


Figure 6: Definition model for Structured ID

Structured IDs forestall collision among themselves by offering upper-side identifiers to the subset of detailed Structured IDs designed by a component provider.

## 5.2 Design of relational links for define meaning

Structured IDs are just techniques to identify data, so they are not able to solve (1) deficiencies in relations of meaning or (2) deficiencies in relations of networking.

Since we define Structured IDs easily, we denote a Structured ID by writing down its identifier of elements, separated by “/” slashes as in Figure 4 and then manage those Structured IDs on the basis of tree structures.

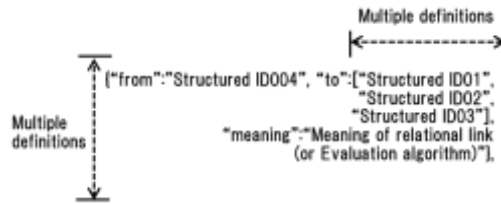


Figure7: Definition model for relational link.

This means that we cannot define a “loop network structure” and/or “cross-coupled network structure.” Therefore, a flexible method is required for defining the relationships of networks. For these two problems, we propose a relational link as way to define relations of subsets of Structured IDs flexibly (see Figure 7).

### 5.2.1 Defining meaning by using relational links

Relational links can be utilized to resolve lack of meaning and relevance of the definition data. For example, if a unique Structured ID defined by the component provider means the failure of the facility, those relations can be defined by using the relational link from a unique Structured ID defined by the component provider to an abstract Structured ID defined by the Sler. There are five types of relational links.

### 5.2.2 One-to-one relationship link

The one-to-one relational link is defined in Figure 8.

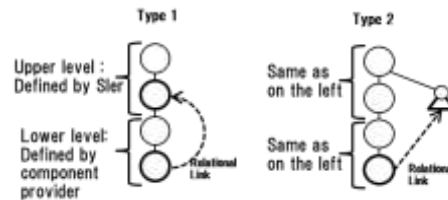


Figure8: One-to-one relational link.

Type 1 shows relations using an arrow with a broken line that includes two Structured IDs. This type of definition is used when an operator-associated Structured ID is defined by a component provider and an abstract Structured ID is defined by a Sler. In another example, Type 1 can define the relationship between a) the Structured ID and b) the dependability policy defined by the Sler. Type 2 is a relationship that connects beyond the parent-child relationship.

### 5.2.3 One-to-multi relational link

The one-to-multi relational link is defined in Figure 9.

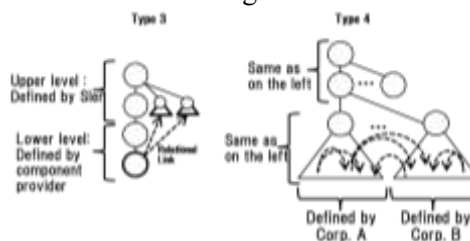


Figure9: One-to-multi relational link.

As a one-to-multi relational link, type 3 can define a relation in which one Structured ID defined by the component provider means multiple Structured IDs defined by an SIER. Type 3 also shows the relationship of the physical network and the physical machine to the virtual network and virtual machine.

Type 4 is a relationship representing the network connection.

#### 5.2.4 Multi-to-one relational link

The multi-to-one relational link is defined in Figure 10.

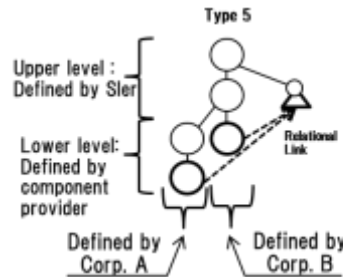


Figure10: Multi-to-one relational link.

As a multi-to-one relational link, type 5 can define relations between virtual sensor technology that utilizes an IIoT sensor and the relational link of failure forecasting.

In a multi-to-one relational link, the approach for Structured ID that meets the three requirements can define a complex control system structure. On the other hand, a multi-to-multi relational link is a composite type that consists of types 1 to 5.

Because of its flexibility, a relational link is able to add knowledge layers if the operator wants to obtain that knowledge by using several relational links online.

The more layers of relational links the control system management software has, the higher the evaluation quality. All Structured IDs in relational links are always a subset of the whole Structured ID set, so control system management software can comprehensively access data by using the Structured IDs.

## 6 Information retrieval for Structured IDs

To address problems of the filtering and retrieval techniques mentioned in section 2.2, we first discuss the single-dimensional space model in Structured ID retrieval and then describe a multi-dimensional (vector space) model in Structured ID retrieval.

### 6.1 Single-dimensional space model in Structured ID retrieval

To manage consistent identifiers, we set the three abovementioned requirements on Structured IDs. These requirements force the retrieval strategy to return only the information that is sought. By comparing each Structured ID, we have thus far viewed a Structured ID as a full sequential string path. This means that a Structured ID is given a set as points in single-dimensional space. In this situation, the information retrieval function and each stakeholder (see Figure 3) connected to the chatroom-style real-time messaging platform use information of "Compare target" and "Compare rule" (see Table 1).



Table 1: Comparison Information in single-dimensional retrieval for Structured ID

Compare target	Compare rule
Structured ID ( $C_t$ )	Structured ID ( $C_r$ )

$$\text{Retrieval function}(C_t, C_r) = \begin{cases} 1 & \text{if } C_t \text{ completely mach } C_r \\ 0 & \text{if } C_t \text{ does not completely mach } C_r \end{cases} \quad (1)$$

Almost all SCADA systems must display detected information on each facility map to compare target Structured ID ( $C_t$ ) and Structured ID ( $C_r$ ) on the basis of formula (1).

## 6.2 Multidimensional space model in Structured ID retrieval

In machine learning, as we develop the idea of weighing the importance of a Structured ID in a message, we might need to filter messages on the basis of statistics of occurrence of the Structured ID. We discuss relaxing the requirement of matching specific regions of a Structured ID; this means we design extended queries that simply consist of query terms with specifications on their relative order, their importance, or their position in a Structure ID.

In this case, we need to transform a Structured ID in a chatroom-style real-time messaging platform into points in a multidimensional space and then search for the target Structured ID using an extended query. To this end, in Structured ID retrieval, we must separate the subset of the Structured ID. One way of doing this is to have each dimension of the vector space encode a word together with its position within the Structured ID tree. A Structured ID contains at least one ID, so we can take each ID string and break it into multiple IDs by using “/” as a separator (see Figure 11).

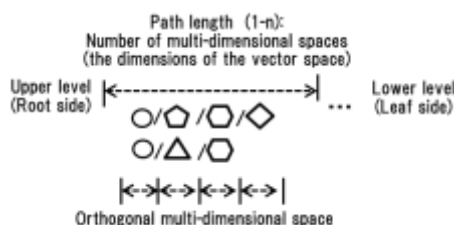


Figure11: Orthogonal multidimensional space in Structured ID

We can also design an extended query in which all parent-child relationships in queries are interpreted as descendant relationships with any number of intervening IDs allowed. For example, we can make an extended query corresponding to the whole Structured ID with the leaf-side node replacing certain words with "\*" as a wildcard (see Table 2). The regular expressions are composed of two types of characters. Specifically, "\*" is special characters called metacharacters, while the rest are called literal, or normal text, characters. In some programming languages, the metasequence to match an actual slash is a slash preceded by a backslash.

Table 2: Extended query notations as regular expression.

Unconsidered Dimensions	Extended Query Notations as Regular Expressions	
Leaf-side multidimensions in Structured ID	RootID/ID1/*	
Middle area multidimensions in Structured ID	RootID/ID1/*/IDn	
Middle area multidimensions in Structured ID	Information:	RootID/DomainID*/I-[0-9]+
(with specific string)	Warning:	RootID/DomainID*/W-[0-9]+
	Fault, Error, Failure:	RootID/DomainID*/E-[0-9]+

The information retrieval function and each stakeholder (see Figure 3) connected to the cha-troom-style real-time messaging platform use information of "Compare target" and "Extended query" (see Table 3).

Table 3: Using Extended Query in Multidimensional Retrieval for Structured ID

Compare target	Extended query
Structured ID ( $C_t$ )	Regular Expression as query about Structured ID ( $C_f$ )

$$\text{Retrieval function}(C_t, C_f) = \begin{cases} 1 & \text{if } C_t \text{ mach } C_f \\ 0 & \text{if } C_t \text{ does not mach } C_f \end{cases} \quad (2)$$

The message filtering system between each platform can use extended query ( $C_f$ ) to filter the target Structured ID ( $C_t$ ) on the basis of formula (2).

It is convenient for machine learning to be able to issue such extended queries without having to specify the exact structural configuration in which a query term should occur because those machine learning functions either do not consider the exact configuration or know enough about the schema of the collection to be able to specify it.

## 7 Illustration of Structured IDs and relational links

In this chapter, we illustrate Structured ID and relational links.

### 7.1 Illustration of Structured IDs

In this section, we illustrate Structured IDs on the basis of the design described in section 5.1. Structured IDs have a list structure from left to right. The longer the Structured ID, the more

information it has for retrieval. As a result, the control system management software can find the target information. Table 4 shows Structured IDs for the wind power generator in Figure 1. If we define a Structured ID on the basis of its proposed design, we use a machine and human friendly ID. Table 1 has limited space, so we use “#” and a number for each ID.

Table 4: Structured IDs for wind power generator (ex. Figure 1)

Each ID	Management Target (meaning)	Structure ID
#1	All System (ID for Whole Management Target)	#1
#2-1	Japan	#1/#2-1
#3	Plant-A (ID for Facility)	#1/#2-1/#3
#4	Subsystem by A Corp.	#1/#2-1/#3/#4
#5	PC as relay	#1/#2-1/#3/#4/#5
#5-1	Software for communication	#1/#2-1/#3/#4/#5/#5-1
#5-2	Hardware	#1/#2-1/#3/#4/#5/#5-2
#6	DB server	#1/#2-1/#3/#4/#6
#7	PLC	#1/#2-1/#3/#4/#7
#8	Wind power generator	#1/#2-1/#3/#4/#8
#9	Subsystem by Corp. B.	#1/#2-1/#3/#9
#5	Personal computer as relay	#1/#2-1/#3/#9/#5
#5-3	Software for communication	#1/#2-1/#3/#9/#5/#5-3
#5-2	Hardware	#1/#2-1/#3/#9/#5/#5-2
#6	DB server	#1/#2-1/#3/#9/#6
#10	Transformer	#1/#2-1/#3/#9/#10
#11	Circuit breaker	#1/#2-1/#3/#9/#11
#12	Protection relay for power plant	#1/#2-1/#3/#9/#12

#13	Failure ID in component by Corp. A.	#1/#2-1/#3/#9/#12/ #13
#13	Failure ID in software by Corp. A.	#1/#2-1/#3/#4/#5/#5-1/#13
#14	Serial communication line	#1/#2-1/#3/#4/#14
#15	Server network by A Corp.	#1/#2-1/#3/#4/#15
#16	Dedicated communication line	#1/#2-1/#3/#9/#16
#17	Server network by Corp. B.	#1/#2-1/#3/#9/#17

Structured IDs #1~#12 are defined by a SIer, and Structured IDs #13 and above are defined by a component provider. Examples are subsystem and failure IDs that only the component provider knows. The IDs of “PC as relay (#5)” cause collisions. However, offering upper-side identifiers forestalls collisions among Structured IDs. In another case, there is a collision between “Failure ID in component by Corp. A (#13)” and “Failure ID in software by Corp. A (#13).” Those IDs have different meanings but the same ID. In this case, offering the upper-side identifiers forestalls collisions among Structured IDs.

As described above, the approach for Structured IDs that meets the three requirements can define a complex control system.

## 7.2 Illustration of relational links

In this section, we illustrate relational links on the basis of the concept in section 5.2.

Table 5: Relational links for wind power generator (ex. Figure 1).

#	Management target (meaning)	Relational links
1	Associating a failure ID defined by Corp. B. to “failure of whole Protection Relay”	{“from”: “#1/#2-1/#3/# 9/ #12/ #13”, “to”:[“#1/#2-1/#3/# 9/ #12”, “#1/#2-1/#3/# 9”], “meaning”:“Relation to failure”}
2	Definition of dedicated communication line “#1/#2-1/#3/ #9/ #16” (Define directions of communication lines.)	{“from”:#1/#2-1/#3/# 9/#12”, “to”:[“#1/#2-1/#3/#9/#5”], “meaning”:“Network”}, {“from”:#1/#2-1/#3/#9/#5”, “to”:[“#1/#2-1/#3/# 9/ #12”], “meaning”:“Network”}

Relational links can define the meaning of a subtree in Structured IDs by using vector notation among Structured IDs (see Table 5). Table 5 details relational links that can solve the two problems in section 5.2: deficiencies in relations of meaning and deficiencies in relations of networking. #1 in Table 5 denotes meaning about relations of failure IDs, and #2 denotes relations of networks.

## 8 Evaluation of proposed techniques in industrial system

In this chapter, we evaluate the attachability of Structured IDs, feasibility of relational links, and deployment of both proposed methods. In addition, we evaluate a filtering technique using information retrieval by regular expressions.

### 8.1 Evaluation Structured IDs and relational links

In this section, we verify that the Structured IDs and relational link can be implemented. The Sler, server vendor, and network vendor can design each Structured ID as their knowledge (see Figure 12). Since those Structured IDs are based on requirement 3 (attachability), they can be combined (see Figure 13).

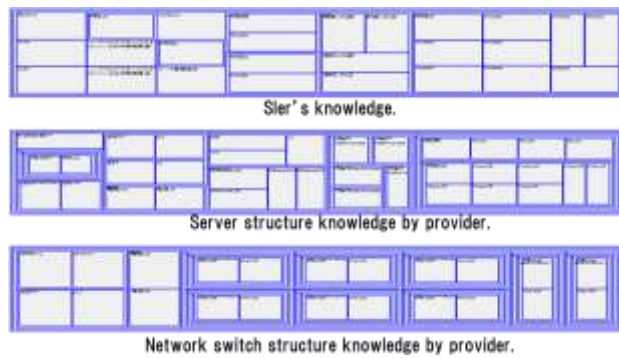


Figure12: Each element of Structured ID



Figure13: Structured ID can be attached on basis of requirement 3 (attachability)

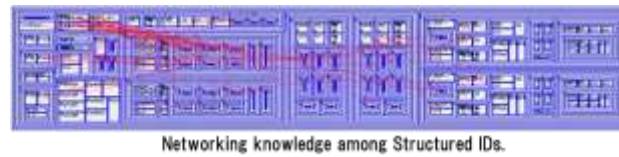


Figure14: Relational link data on computer system

We can chose Structured IDs to make relational links about network relations (Figure 14).

### 8.2 Development IIoT SCADA using proposed techniques

To evaluate the application of the proposed methods on a real system, we program IIoT SCADA using web functions such as HTML5, node.js, and socket.io. We use a Structured ID as an identifier for all components, such as the information output source, each function, and views for various stakeholders.

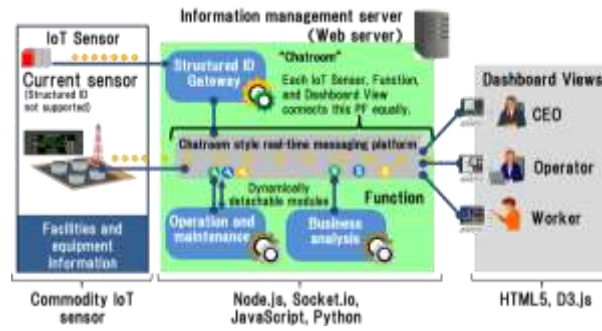


Figure15: Evaluated implementation of IIoT SCADA.

In a chatroom-style real-time messaging platform, all connections from IIoT sensors, knowledge management functions, views, etc. must be treated equally without bias. This integrated transaction platform requires every ID to be unique. Thus, when we use this platform, we use Structured IDs (see Figure 16).



Figure16: Implemented views for (a) worker and (b) Operator.

### 8.3 Evaluation filtering function using orthogonal Structured ID retrieval

In this section, we verify that orthogonal Structured ID retrieval by regular expressions.

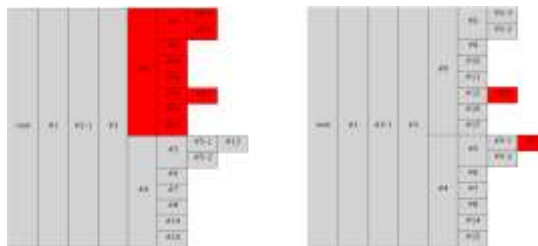


Figure17: Evaluation of extended query by regular expression  
 (a) Leaf-side multidimension filtering,  
 (b) Middle area multidimension path through filtering.

In the evaluation of filtering, we send all structured IDs of the target system and then check results of filtering on a web browser. If the result is the same as the logical result, the function is functioning properly. Leaf-side multidimensional filtering output the correct result (see Figure 17 (a), left side). We find a partial area that was turned red.

Middle area multidimension path through filtering output the correct result (see Figure 17 (b), right side). We find each partial area that was turned red in different branches.

Finally, we demonstrated that we can integrate the IIoT SCADA system by dynamic attachable

modules (see Figure 18).

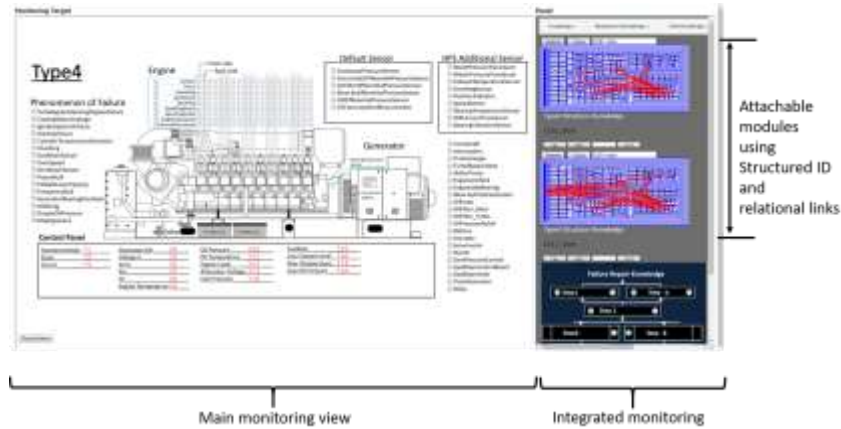


Figure18: Integrated IIoT SCADA using proposed techniques with dynamic attachable modules for predictive maintenance

## 9 Conclusion

The Industrial Internet of Things (IIoT) concept is causing problems in the management of consistent identifiers. To solve these problems, we set three requirements: (1) Uniqueness, (2) Concreteness, and (3) Attachability. We also proposed Structured IDs, which can define a complicated structure of control systems using IDs on a tree structure. Additionally, we proposed relational links, which can describe meaning for subsets of Structured IDs. Furthermore, in use cases in which consistently managed information is utilized, a method of distributing appropriate information to each stakeholder is required. Additionally machine learning functions do not consider the exact configuration or know enough about the schema of the collection to be able to specify it. To address this requirement, we propose an orthogonal information retrieval technique for Structured IDs. Finally, we verified that the proposed techniques can be implemented using web technology and demonstrated that they can solve problems in an IIoT use case. Filtering is a real problem that needs to be solved with many types of programming. However, programming with regular expressions and structured IDs can make the job substantially easier.

## Acknowledgement

We would like to express my sincere gratitude to emeritus professor Toshinori WATANABE, Dr. Shoichi MASUI, Dr. Ryuichiro KODAMA, Dr. Sadahiro ISHIKAWA, Dr. Toshihiko NAKANO, Dr. Motohisa FUNABASHI, Mr. Toshiaki KOBARI, Mr. Shinichi SHIOBARA, Mr. Tadashi SUZUKI, Mr. Munechika KOTANI, Mr. Naoki MURAKAMI, and Mr. Takeshi WATANABE for providing us this precious study opportunity.

## References

- [1] D. N. P Thalakotuna, T. A. T Tilakaratne, Y. B. M. S Jayasinghe, T. H. K. M Silva, and S. R. Munasinghe, "Remote Monitoring and controlling system for a water purification plant," 2008 4th International Conference on Information and Automation for Sustainability, ISBN. 978-1-4244-2899-1, 2008, pp. 372 - 377.

- [2] Tao Sun, Bing Zhao and Fanbing Li, "Application of WinCC in Carplant Monitoring System," Computer Science and Society, 2011 International Symposium, ISBN. 978-1-4577-0644-8, 2011, pp. 203 - 205.
- [3] B. Stojkovic and M. Vukasovic, "A new SCADA System design in the Power System of Montenegro," ICCP/TASE.2 and Web-based real-time electricity demand metering extensions. 2006 IEEE PES Power Systems Conference and Exposition, ISBN. 1-4244-0177-1, 2006, pp. 2194 - 2199.
- [4] W. L. Xia, Z. Wang, Y. R. Lu, J. Zhao, J. Chen, H. T. Ren, S. X. Peng, and J. E. Chen, "Design and Implementation of a Compact Control System for Coupled RFQ-SFRFQ Linac," IEEE Transactions on Nuclear Science, Volume:61 , Issue: 4, 2014 pp. 2345 - 2350.
- [5] WEN Jie, DAI Jiao, HAN Jizhong, DI Fangchun, and YAN Hu, "Llama Distributed Multi-Versioned Model Data Management for SCADA," 2014 IEEE 17th International Conference on Computational Science and Engineering, ISBN. 978-1-4799-7980-6, 2014, pp. 1968 - 1973.
- [6] Tingli LI, Yang LIU, Ye TIAN, Shuo SHEN, and Wei MAO, "A Storage Solution for Massive IoT Data Based on NoSQL," Green Computing and Communications, 2012 IEEE International Conference, ISBN. 978-1-4673-5146-1, 2012, pp. 50 - 57.
- [7] Wei-Chen Chen, Chao-Lin Wu, Ya-Hung Chen, and Li-Chen Fu, "An Efficient Data Storage Method of NoSQL Database for HEM Mobile Applications in IoT," Internet of Things, 2014 IEEE International Conference on, and Green Computing and Communications, IEEE and Cyber, Physical and Social Computing, ISBN. 978-1-4799-5967-9, 2014, pp. 336 - 339.
- [8] Phan Thi Anh Mai, Jukka K. Nurminen, and Mario Di Francesco, "Cloud Databases for Internet-of-Things Data. Internet of Things," 2014 IEEE International Conference on, and Green Computing and Communications, IEEE and Cyber, Physical and Social Computing, ISBN. 978-1-4799-5967-9, 2014, pp. 117 - 124.
- [9] Tuan Dinh Le, Seong Hoon Kim, Minh Hoang Nguyen, Daeyoung Kim, Seung Young Shin, Kyung Eun Lee, and Rodrigo da Rosa Righi, "EPC Information Services with No-SQL datastore for the Internet of Things," 2014 IEEE International Conference on RFID, ISSN. 2374-0221, 2014, pp. 47 - 54.
- [10] Marco Mesiti and Stefano Valtolina, "To-towards a User-Friendly Loading System for the Analysis of Big Data in the Internet of Things," Computer Software and Applications Conference Workshops, 2014 IEEE 38th International, INSPEC Accession Number. 14614879, 2014, pp. 312 - 317.
- [11] Kunal Taneja, Qian Zhu, Desmond Duggan, and Teresa Tung, "Linked Enterprise Data Model and its use in Real Time Analytics and Context-Driven Data Discovery," 2015 IEEE International Conference on Mobile Services, ISBN. 978-1-4673-7283-1, 2015, pp. 277 - 283.
- [12] Davy Preuveneers and Yolande Berbers, "SAMURAI: A Streaming Multi-tenant Context-Management Architecture for Intelligent and Scalable Internet of Things Applications," Intelligent Environments, 2014 International Conference, INSPEC Accession Number.



14618663, 2014, pp. 226 - 233.

- [13] Minbo Li, Zhu Zhu, and Guangyu Chen, "A Scalable and High-efficiency Discovery Service In IoT Using a New Storage Schema," Computer Software and Applications Conference, 2013 IEEE 37th Annual, INSPEC Accession Number. 13879461, 2013, pp. 754 - 759.
- [14] Martín Díaz, Gonzalo Juan, Oikawa Lu-cas, and Alberto Ryuga, "Big Data on the Internet of Things. Innovative Mobile and Internet Services in Ubiquitous Computing," 2012 Sixth International Conference, ISBN. 978-1-4673-1328-5, 2012, pp. 898 - 900.
- [15] Liang Jeff Chen, Philip A. Bernstein, Peter Carlin, Dimitrije Filipovic, Michael Rys, Nikita Shamgunov, James F. Terwilliger, Mi-los Todic, Sasa Tomasevic, and Dragan Tomic, "Mapping XML to a Wide Sparse Table," IEEE Transactions on Knowledge and Data Engineering, Volume:26 , Issue: 6, INSPEC Accession Number. 14349418, 2014, pp. 1400 - 1414.
- [16] W3C : Extensible Markup Language (XML), <https://www.w3.org/XML/> , Apr. 2017.
- [17] W3C : XML Path Language (XPath), <https://www.w3.org/TR/xpath/> , Apr. 2017.