International Journal of Informatics Society

17/12 Vol. 9 No.3 ISSN 1883-4566

Informatics Society

Aims and Scope

The purpose of this journal is to provide an open forum to publish high quality research papers in the areas of informatics and related fields to promote the exchange of research ideas, experiences and results.

Informatics is the systematic study of Information and the application of research methods to study Information systems and services. It deals primarily with human aspects of information, such as its qu ality and value as a resource. Informatics also referred to as Information science, studies t he structure, algorithms, behavior, and interactions of natural and a rtificial systems that store, process, access and communicate information. It also develops its own conceptual and theoretical foundations and utilizes foundations developed in other fields. The advent of computers, its ubiquity and ease to use has led to th e study of info rmatics that has computational, cognitive and social aspects, including study of the social impact of information technologies.

The characteristic of informatics' context is amalgamation of technologies. For creating an informatics product, it is necessary to integrate many technologies, such as mathematics, linguistics, engineering and other emerging new fields.

Guest Editor’s Message

Yoshitaka Nakamura

Guest Editor of Twenty-Seventh Issue of International Journal of Informatics Society

We are delighted to have the twenty-seventh issue of the International Journal of Informatics Society (IJIS) published. This issue includes selected papers from the Tenth International Workshop on Informatics (IWIN2016), which was held at Riga, Latvia, Aug. 28-31, 2016. The workshop was the tenth event for the Informatics Society, and was intended to bring together researchers and practitioners to share and exchange their experiences, discuss challenges and present original ideas in all aspects of informatics and computer networks. In the workshop 26 papers were presented in eight technical sessions. The workshop was successfully finished with precious experiences provided to the participants. It highlighted the latest research results in the area of informatics and its applications that include networking, mobile ubiquitous systems, data analytics, business systems, education systems, design methodology, intelligent systems, groupware and social systems.

Each paper submitted IWIN2016 was reviewed in terms of technical content, scientific rigor, novelty, originality and quality of presentation by at least two reviewers. Through those reviews 15 papers were selected for publication candidates of IJIS Journal, and they were further reviewed as a Journal paper. This volume includes five papers among the accepted papers, which have been improved through the workshop discussion and the reviewers’ comments.

We publish the journal in print as well as in an electronic form over the Internet. We hope that the issue would be of interest to many researchers as well as engineers and practitioners over the world.

Yoshitaka Nakamura received the Bachelor of Engineering Degree in Informatics and Mathematical Science and the Master of Information Science and Technology Degree from Osaka University in 2002 and 2004, respectively. He also received Doctor of Philosophy in Information Science and Technology Degree from Osaka University in March 2007. He was a research fellow of the Japan Society for the Promotion of Science (JSPS) from April 2006 to March 2007. From April 2007 to March 2009, he was an assistant professor of Nara Institute of Science and Technology (NAIST), and from April 2010 to March 2011, he was a specially appointed assistant professor of Osaka University. And from April 2011 to March 2016, he was a research associate in Future University Hakodate. He is now an associate professor in the School of Systems Information Science, Future University Hakodate. His research interests include information security, mobile network and ubiquitous computing. He is a member of the Information Processing Society of Japan (IPSJ), the Institute of Electronics, Information and Communication Engineers (IEICE), and the Institute of Electrical and Electronics Engineers, Inc. (IEEE).

An Application of MongoDB to Enterprise System Manipulating Enormous Data

Tsukasa Kudo[†], Yuki Ito[†], and Yuki Serizawa[†]

[†]Faculty of Comprehensive Informatics, Shizuoka Institute of Science and Technology, Japan
kudo@cs.sist.ac.jp

Abstract - With the spread of the IoT, a variety of sensor data have been widely used, such as the image data of surveillance cameras and so on. And, such a system operations are spreading, in which the image data is saved in the database directly. Here, the relational database management system has a problem about the efficiency to manipulate the enormous unstructured data such as images. So, it is spreading that the system manipulating such a data is implemented by using the NoSQL databases such as the MongoDB with GridFS interface. However, as for the enterprise system, since the ACID properties of the transaction cannot be maintained in MongoDB, there has been the problem about its application to such a system. On the other hand, in our previous study, we had implemented the transaction feature for MongoDB to maintain all the ACID properties. Therefore, it is expected that MongoDB can be applied to such a system by using this transaction feature. That is, the advantage of MongoDB can also be used in the enterprise systems. In this paper, we show the application case of MongoDB to the production management system, which is a kind of enterprise system; also, we show the image management for the stocktaking as the case of the efficiency improvement by using MongoDB.

Keywords: database, transaction processing, ACID properties, MongoDB, GridFS, production management system

1 INTRODUCTION

With the spread of the IoT, networking sensors are collecting various types of data including the semi-structured and unstructured data such as images, audio and videos（hereinafter, “image and video”) [5], [8], [13]. Since these data is stored into the database and shared, it has become necessary that the database management systems adopt the feature called 3V, that is, Volume (huge amount), Velocity (speed) and Variety (wide diversity) [11]. Here, the relational database management systems (RDBMS) was designed for handling structured data, so it has limitations when it comes to manage the enormous and numerous unstructured data like files [19]. Thus, the various database management systems called NoSQL database, which is different from the conventional RDBMS, have been proposed and put to practical use [20]. MongoDB is a kind of NoSQL database and equips GridFS interface [1], by which various types of enormous data are stored efficiently by dividing into units called chunk, and it has been proposed and evaluated to apply to the above-mentioned fields [19], [26].

By the way, our laboratory is supporting the introduction of the production management system for the actual company. At its factory, the products are manufactured by the order-made, and their parts are replenished in lot unit only when they are insufficient. However, since the type of parts are so many in this factory, their accurate inventories often cannot be grasped. As a result, it has become the factor causing the shortage of the parts.

For this problem, we converted the idea from the inventory management by counting the actual quantity. That is, the human vision can grasp the approximate number of the actual inventory efficiently. So, we conceived that if the necessary inventory quantity is designated, the worker can find that the inventory is insufficient by looking at the inventory shelf. And, MongoDB can be applied to treat the image and video data that is the evidence of his judgment. Here, in this production management system, it is also necessary to manage the theoretical inventory, which is the quantity of the parts inventory managed in the system. So, in the case of moving 10 parts from the parts shelf to the assembly field, 10 is subtract from the former *collections* (corresponding the table in RDBMS) of the database; the same number is added to the latter *collections*. And, to maintain the consistency, these two data operations must be performed as a single transaction.

However, as for MongoDB, the ACID properties are maintained only in the case of updating a single data. So, in the case of updating the plural data, the data is updated one after another, and finally they become to be updated. That is, there is the anomaly in the midst of this updating, such as one data has been updated and another has not been updated; and, it can be queried by the other transactions. So, we could not find the application case of GridFS interface of MongoDB to the enterprise systems, though it provides the useful feature for the enormous data manipulations.

On the other hand, as for this problem, we showed the method of transaction feature for MongoDB for the centralized database environment [10]. By this method, all the ACID properties, which is consist of the atomicity, isolation, consistency, and durability, can be maintained throughout in each transaction. Moreover, this production management system can be built with the centralized database environment.

Therefore, we decided to apply MongoDB equipping this transaction feature to this production management system, and implemented its prototype. As a result, we confirmed that MongoDB could be applied to the enterprise system by using our transaction feature. Moreover, we confirmed the image and video data management by using MongoDB could improve the efficiency of inventory management works.

The remainder of this paper is organized as follows. Section 2 shows our transaction feature and the problem of the target production management system. And, we show the application method of MongoDB to the enterprise system in Section 3. Section 4 shows the implementation of this method and its evaluation results. We discuss on the results in Section

5, and Section 6 concludes this paper.

2 RELATED WORKS AND PROBLEMS

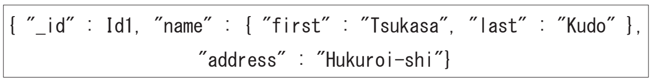
2.1 Transaction Feature for MongoDB

MongoDB is a document-oriented NoSQL database, and its data is stored as a *document* of the JSON (JavaScript Object Notation) format as shown in Fig. 1 [14], [25]. Its *document* is composed of the fields, and {"_id": Id1 } expresses the field having the identifier "_id " and value "Id1". Here, "_id" indicates the ObjectID that corresponds to the primary key of the relational databases. In Fig. 1, the other fields of the *document* are "name" and "address". Here, field "name" has a nested structure, which is composed of field "first" name and "last" name. Since MongoDB has such a structure, it is not necessary to define the scheme of the database beforehand like RDBMS. That is, the fields of each *document* can be added or removed at any time. Incidentally, the set of *documents* composes the *collection*, and each of them corresponds respectively to the records and table in the relational database although not strictly. Thus, each *collection* can have *documents* of various structures.

In addition, similar to SQL of the RDBMS, CRUD (Create, Read, Update, Delete) data manipulation is provided. However, since its transaction feature is based on the eventual consistency of the BASE properties [3], [18], the ACID properties can be maintained as for only the single *document*. Therefore, there is the problem that the ACID properties of transactions cannot be maintained in the case of updating multiple *documents* simultaneously. Incidentally, the ACID properties are defined as the following 4 properties: Atomicity means the transaction updates completely or not at all; Consistency means the consistency of database is maintained after it is updated; Isolation means each transaction is executed without effect on the other transactions executing concurrently; Durability means the update results survive the failure [7].

For example, in the case of the bank account transfer from account A to account B, the total amount of the both accounts does not change. However, as shown in (a) of Fig. 2, since the ACID properties are not maintained on the entire updating in MongoDB, there is the problem of the anomaly. That is, the halfway state during the updating is queried by the other transactions: one data has been already updated; another data has not been updated yet. In this case, when the account A was updated, the anomaly that the sum of the query result was reduced to 2,000 temporarily happened. Also, since the rollback must be executed by the compensation transaction in the case of failure [22], it has been shown that the isolation on these documents cannot be maintained by this method [12]. That is, the isolation cannot be maintained in the both cases of the commit and rollback. On the other hand, in the case where the same procedure was performed in the RDBMS, this halfway state can be concealed until the commit as shown in (b) of Fig. 2.

Furthermore, as for SQL of the RDBMS, the isolation levels of the transactions are defined. That is, corresponding with the business requirement, the suitable isolation level can be selected: in the case where it is needs the efficient execution, or the strict concurrency control [7]. So, we had implemented and evaluated the method to perform each transaction with the designated isolation level in MongoDB in the previous study, as well as the RDBMS as shown in Table 1. As a result, we confirmed the following: the isolation levels of Table 1 were achieved, and the query transaction performance at READ UNCOMMITTED is same as MongoDB [10].



```
{ "_id" : Id1, "name" : { "first" : "Tsukasa", "last" : "Kudo" },
              "address" : "Hukuroi-shi"}
```

Figure 1: An example of MongoDB document.

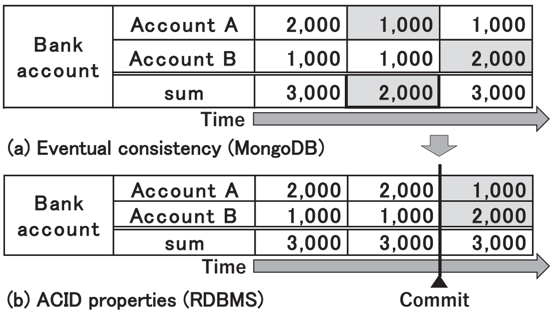


Figure 2: Problem of transaction processing in MongoDB

We show the overview of this method below. First, it performs the lock operation during the access to the *document* as well as the RDBMS as shown in Table 1. In Table 1, "2PL" shows the two phase locking protocol [7]. Incidentally, in this method, to prevent the cascade abort of the transaction, the rigorous 2PL is adopted as well as the RDBMS. That is, the lock is held until the commit or rollback.

Second, as shown in Fig. 3, the *document* of the *Data collection* saves the business data into two fields, "Data before update" and "Data after update". While the *document* is not being updated, the business data is saved in the former; and, there is not the field of the latter. On the contrary, while the *document* is being updated, the business data of two state is saved: the state of before update is saved in the former; and, the state of after update is saved in the latter. In addition, it has the fields to save the information of the transactions locking it: for each of the shared lock and the exclusive lock. And, in order to manage the transactions that are locking the *document* of *Data collection*, we implemented *TP (transaction processing management) collection*. And, its *document* saves the corresponding transaction state: before or after the commit. Also, it saves the isolation level of the transaction.

For the case of Fig. 2, we show the procedure to query the *document* of *Data collection* while it is being updated with the isolation level READ COMMITTED or REPEATABLE READ. If the bank account transfer from the account A to ac-

Table 1: Locking protocol of each isolation level

Isolation level	Exclusive lock	Shared lock
READ UNCOMMITTED	2PL	(none)
READ COMMITTED	2PL	While query
REPEATABLE READ	2PL	2PL

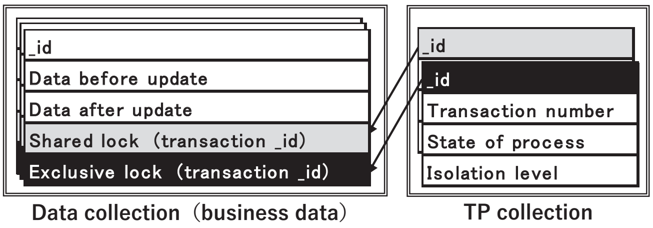


Figure 3: Transaction processing method for MongoDB

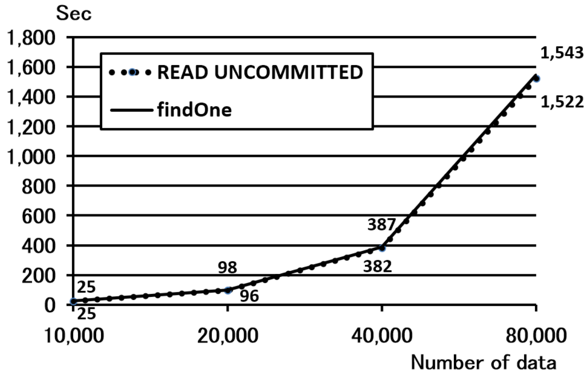


Figure 4: Comparative evaluation with findOne of MongoDB

count B shown in (a) of Fig. 2 is executed by this method, then both of the data before and after update are retained. And, the data before update is queried until the commit of the transaction; the data after update is queried after the commit. Therefore, in (a), both update results of the account A and B are not queried until the commit; both update results are queried only after the commit. That is, the query results are similar to the RDBMS shown in (b). Thus, the halfway state during the updating is concealed from the other transactions, and the transaction processing maintaining the isolation can be provided.

Similarly, other properties of the ACID propertied can be also maintained by this method. Basically, as for the update of individual *document*, the ACID properties are maintained by the transaction feature of MongoDB itself. And, as for the atomicity, the data before update is saved in “Data before update” field. Therefore, when the failure occurs while updating plural data, the rollback of all the *documents* can be performed by deleting their “Data after update” fields. As a result, since the rollback can be performed without using the compensation transactions, the isolation property can be maintained. Incidentally, though the compensation transactions is usually used for the rollback in MongoDB [22], it cannot maintain the isolation. On the contrary, in the case where this update completes normally, their commit can be performed by changing “Data after update” to “Data before update”. Then, the former is deleted. As for the consistency, it also can be maintained by this rollback when the consistency is not maintained during updating plural *documents*.

As for the durability in the event of a crash, it is provided by the original transaction feature of MongoDB, which uses write ahead logging to an on-disk journal [16]. And, since this method uses this transaction feature, the durability can be maintained in this method, too.

On the other hand, the Velocity (speed) of the 3V feature, that is, high efficiency for the query of the NoSQL databases is generally required. So, we performed the comparative evaluation on the query processing between this method and MongoDB. As for the former, the query transaction was performed with the isolation level READ UNCOMMITTED, in which the query was performed efficiently without the shared lock as shown in Table 1. As for the latter, we used “findOne” method, which was the standard query method of MongoDB. As a result, we found that the performance of the both are almost the same as shown in Fig. 4 [10].

Here, it has been shown that if all the transactions are performed with any of the isolation level shown in Table 1, then any transaction can be performed with the designated transaction level [7]. In other words, by using this method, the usual query transactions can be performed efficiently with the isolation level READ UNCOMMITTED as in the conventional data manipulation of MongoDB; only the query and update transactions, which need the strict data manipulations as shown in Fig. 2, can be performed with the isolation level READ COMMITTED or REPEATABLE READ.

2.2 Enormous Data Manipulation Feature in Databases

As for the enormous data manipulation, the RDBMS has some problem: there is the limitation of the data size, and it must be read sequentially. We show this in detail as follows.

As for the relational databases, there had been the request to treat the various type of enormous data including the image and video. So, in SQL:1999, known as SQL3, the data type LOB (LARGE OBJECT) have been defined. It is composed of the data type CLOB (CHARACTER LARGE OBJECT) and BLOB (BINARY LARGE OBJECT): the former treats the character strings; the latter treats the binary data including the image, video and so on [6]. However, it has been shown: since RDBMS was designed for handling structured data, it has limitation to manage the enormous unstructured data [26], [19]. That is, the enormous unstructured data in binary-based column increases the demand for hardware resources, and the distributed systems to reduce this problem tend to be rigid and hard to administrator. And, it has been shown that using RDBMS for managing the enormous unstructured data is inefficient, because it is due to the architecture for the concurrency control by using locking, logging and so on [27], [28].

In addition, for example, MySQL (MySQL 5.7) also supports the BLOB type, and the data up to 4GB can be stored. However, as for the BLOB type, some restrictions have been shown [17]. For example, VARBINARY type is recommended in the case of the small size of data (up to 64KB); the column of BLOB type should be separated to another table for the sake of query processing. And, the data size transferred between the server and client, or among the servers, is limited up to 1GB. So, in the case of using the replication feature, the data size must be up to 1GB; in the case of storing the data of more than 1GB to the database, the data must be divided and stored sequentially. Furthermore, even in the case of querying a portion of the enormous BLOB type data, it is necessary to read sequentially from the beginning.

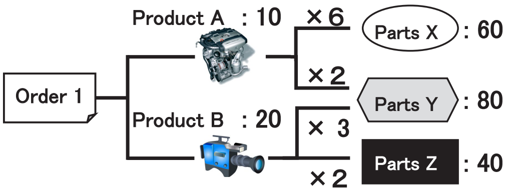


Figure 5: Composition of MRP system

Besides this, several methods are provided to save the unstructured enormous data such as images and videos. For example, Sears et al. showed that the file system has the clear advantage in the case where the data amount was so large [21]. However, in the case where the business system is constructed by the file system, it is pointed out that there are problems: they lack the efficient mechanism for data integration, security, backup and recovery, and so on; the configuration of application software becomes complicated [26], [23].

On the other hand, as for MongoDB, GridFS interface was prescribed, which is the convention to store the enormous data, and the official drivers support this [14]. In this convention, the enormous *document* is divided into chunks as a separated *document*. By using the GridFS, the data which size exceeds the file system of the OS can be manipulated, and the replication is also supported. In addition, as for even the binary data, a portion of the data can be queried. Because of this advantage, MongoDB has been proposed and evaluated in several systems dealing with enormous unstructured data such as images and videos: medical images for health care system, documents for e-learning system and so on [26], [19].

2.3 Target Production Management Business

Our laboratory has been supporting the production management system of a manufacturing company: the implementation, introduction and support of the system operations. Previously, we had introduced the MRP (Material Requirement Planning) system to automate the calculation of the quantity and cost of the parts, which is necessary to assemble the ordered products. We show the over view of the MRP system in Fig. 5. In this case, 2 parts Y is used for product A; 3 for Product B. So, in the case of order 1, which is composed of 10 products A and 20 products B, 80 parts Y is necessary. The cost of the parts, which calculated by this system, is used as the master data of the system in conjunction with the order company by EDI (Electronic Data Interchange) [9].

Now, we have been asked to introduce the inventory management system. The inventory control is the important function of the production management, and it aims to maintain the inventory quantity at the proper levels. In other words, the inventory levels of all the parts should be controlled such that the following can be achieved: the quantities of each parts are always more than the safety inventory, by which some problems can be dealt with to prevent the parts shortage; on the other hand, there should not be too much excess inventory, which causes the increase of the production cost. Here, this company manufactures the products by the order made, and the parts are replenished in lot unit only when they are short.

We show the inventory and flow of parts i in the factory in Fig. 6. Unused parts are kept in “Parts shelf”, then they are delivered to “Assembly field” by “Parts delivery” to manufacture the products. And the finished products are shipped for each order. Here, I_i is the inventory quantity, which is indicated by the dashed box. And, S_i is the safety inventory quantity included in it. Similarly, A_{i1} is the already assigned inventory quantity to the other ordered products in “Parts shelf”; A_{i2} is the one in “Assembly field”, that is, the parts are in process. When the factory receives the new order, the necessary parts quantity R_i is calculated by the MRP system. Then, insufficient quantity P_i is replenished in lot unit. P_i and the assigned total quantity A_i are expressed as follows.

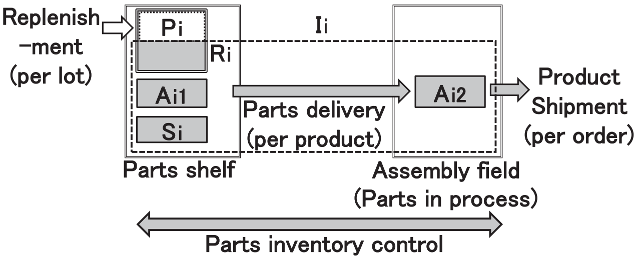

Replenish -ment (per lot)
Pi
Ri
Ii
Ai1
Si
Parts delivery (per product)
Ai2
Product Shipment (per order)
Parts shelf
Assembly field (Parts in process)
Parts inventory control

Figure 6: Product manufacturing process

$$P_i = R_i - I_i + A_i + S_i \tag{1}$$

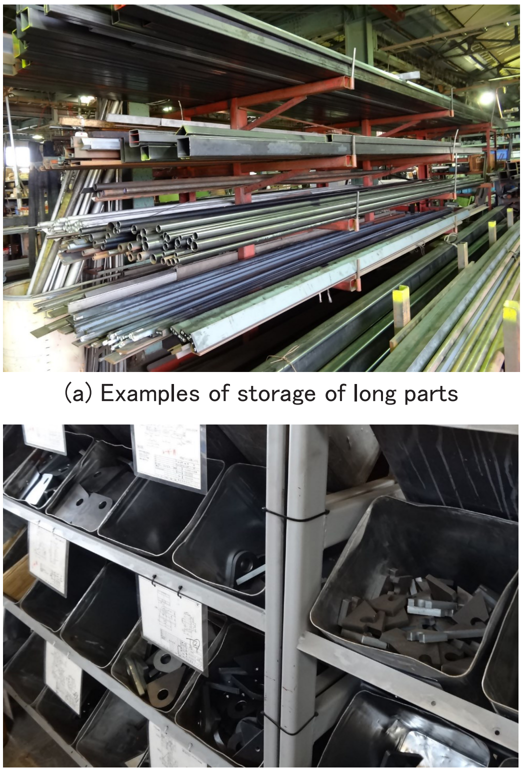
$$A_i = A_{i1} + A_{i2} \tag{2}$$

For example, in the case of Fig. 5, as for parts Y, if R_Y is 80, I_Y is 50, A_Y is 30 and S_Y is 10, then the production quantity P_Y becomes 70. In the factory, replenishment is done in lot units as above-mentioned. So, there is often surplus, and $P_i = 0$ in this case.

Here, it is necessary to grasp the accurate inventory quantity to determine P_i. However, it is not easy in the actual factory. The types of the parts are several hundreds, and the parts shelf are dispersed in various places of the factory to adapt to the individual work process. Figure 7 shows the parts shelf examples as for the long parts and small parts. The long parts have to be counted from a particular direction. And, the small parts are stored in containers. So, it is necessary to take out them in order to count the exact quantities. In this way, it takes time to move among the parts shelf and to investigate the quantities. Actually, it was estimated to take a few man-days for the stocktaking of all the parts. Moreover, the parts are always moving from the parts shelves to the assembly fields, so the actual inventory quantities of parts shelves are always changing, too.

Incidentally, in the field of the production management in the large companies, the large scale production management system is introduced, such as the SAP [4], and the production information is managed as the integrated system including the inventory, accounting, order and so on. Also, the inventory quantity is sometimes measured by using the RFID (radio frequency identifier) tags in the various field to reduce the inventory investigation workload [2].

However, the target factory is the small or medium-sized company like most companies in Japan, of which proportion is said 99.7% [24], and it is pointed out that the introduction of such a management system is so less than the large company.


(a) Examples of storage of long parts

(b) Examples of storage of small parts

Figure 7: Parts shelf in target factory

As for this cause, two factors can be pointed out from the view point of their production scale. First, it is difficult to obtain the effect commensurate with the system investment such as the RFID and so on. Second, it is difficult to reserve the full-time personnel for the system operations, grasping the field data and entering it into the system. However, with the development of the e-commerce and supply chain management (SCM), it is becoming necessary to introduce the EDI with the large companies. Therefore, it is also becoming necessary to introduce the production management system to manage the data for the EDI. And, the inexpensive packages of the production management systems are distributed. However, they need to enter accurate inventory quantities.

As a result, to grasp the actual inventory efficiently and to determine the part production quantity P_i became the requirement of the target inventory management system. And, there are also following supplemental requirements. First, from the viewpoint of the cost performance, the target system must be implemented without using expensive equipments and devices. Second, the system operations must be performed without increasing the workload of personnel.

3 APPLICATION METHOD OF MONGODB

3.1 Novel Method to Grasp Actual Inventory

For the problem mentioned in Section 2.3, we changed our idea about the judge method of parts sufficiency from counting the actual inventory. And, we proposed a method to judge whether the necessary quantity is satisfied or not by the human vision based on the following facts. Firstly, as for the parts delivery or the product shipment, the quantity of each necessary parts can be grasped easily by using the MRP system. That is, it can be calculated automatically by the order ID and necessary quantity of each product, and these data is received via EDI as the electronic data and can be used efficiently. Secondly, the human vision can grasp the approximate number of parts efficiently in the various situations.

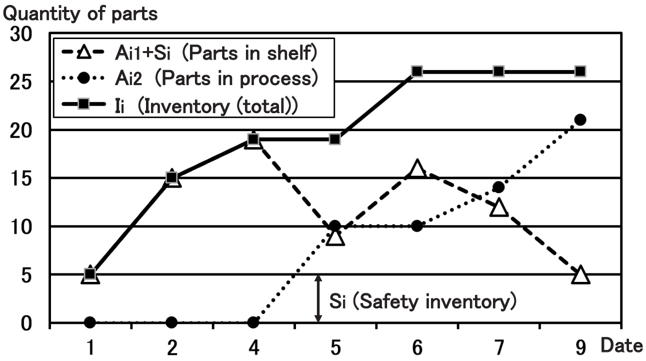


Figure 8: Change of quantity of parts inventory and delivery

Figure 8 shows the change of the theoretical inventory of a parts, which is the necessary quantities and corresponds to equation (1). Incidentally, the product shipments are omitted for the sake of simplicity in this figure. In this case, the parts are prepared in the parts shelf prior to assembly start 3 days, and the safety inventory quantity is 5. For example, 10 parts are prepared (15 including the safety inventory; "$A_{i1} + S_i$") on second, and they are moved to the assembly field on fifth ("A_{i2}"). Similarly, the parts are prepared 4 on fourth; 7 on sixth, and they are delivered on seventh and ninth respectively. So, on second, R_i is 4; I_i is 15; A_{i1} is 10; S_i is 5. Then, the production quantity P_i is 4 on fourth.

However, in the actual field, since there are manufacturing loss and the process delay, they do not always equal to the actual inventory. Therefore, as above-mentioned, to perform the inventory control, the actual inventory must be grasped, too. And, only the judgement that there is the necessary quantity of parts in the parts shelf on the designated date is performed in our proposed method, so it can be done efficiently. For example, assuming that Fig. 8 shows the transition of parts stocked in the rightmost container of Fig. 7 (b), and it is sufficient if more than 15 parts are present on the second day. If we see in Fig. 7 (b), then we find it is easy to judge it by the human vision, even if we do not know the exact inventory quantity. Therefore, by this method, the inventory manager can perform his business efficiently in the office by using the image and video, and he needs no field work. And, in the case where some actual parts inventory may be insufficient, the parts replenishment in lot unit is ordered by the manager.

We show the composition of the proposal system in Fig. 9. The worker takes out the parts from the parts shelf, then take the image and video of this shelf by his hand-held camera. And, he enters it into the database with the parts data: the order ID and product ID. Then, the system calculates the both theoretical inventory as shown in Fig. 8: the one was remained in the parts shelf; the other was delivered to the as-

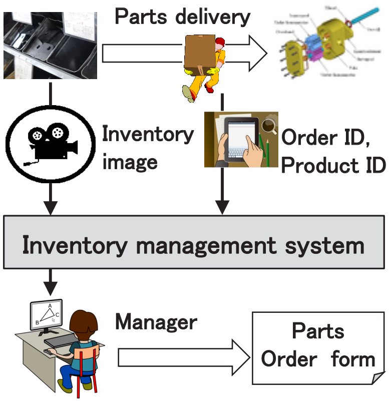


Figure 9: Composition of proposal system

sembly field. In this operation, for example, as shown on the fifth day of Fig. 8, the delivered quantity (here, 10) is reduced from the *document* ($A_{i1} + S_i$) of the parts shelf and the same quantity is added to the *document* (A_{i2}) of the assembly field. And, the total quantity (I_i) that is the sum of the both must not change during this operation.

Here, in order to manage the actual inventory by this method, it is necessary to grasp the exact theoretical inventory shown in Fig. 8. And, since the parts are stocked separately in both the parts shelf and the assembly field, the theoretical inventory is managed by each corresponding *document*. This indicates that the updating of these two *collections* must be processed as a transaction maintaining the ACID properties.

3.2 Requirements for Database Application

As shown in Section 3.1, the database of the proposal system must satisfy the requirements in two sides. The one is the enormous data manipulation to grasp the actual inventory, which is provided by MongoDB as shown in Section 2.2; the other is the transaction management to calculate the theoretical inventory, which is provided by the RDBMS for the usual enterprise system.

That is, the database of the proposal system has to treat not only the character and numerical data of the inventory information, but also the image and video data in the factory. So, if this system was implemented by using the RDBMS, the significant restrictions occurs on the data manipulation as shown in Section 2.2.

For this reason, we used MongoDB for the database of this system. On the other hand, if this system was implemented by using MongoDB, the following restrictions are considered: the ACID properties is not maintained as the whole transaction; the join operation to connect plural *collections* each other is not supported.

In summary, the requirements for the application of MongoDB to the proposal system is as the following. The first requirement is that its transaction can manipulate the plural *collections* with maintaining the ACID properties. The second requirement is its *collections* can be connected each other using only the reference among them. That is, it is composed without the feature not provided in MongoDB: the join operation and so on. The third requirement is that its transaction can efficiently manipulate the enormous data such as images and videos. That is, its manipulation time of such a data must be shorter than the time of MySQL.

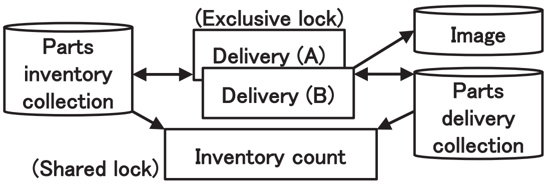


Figure 10: Inventory management model for MongoDB

3.3 Inventory Management Model for MongoDB

As for the first requirement, it could be satisfied by the transaction feature of MongoDB which was the result of our previous study shown in Section 2.1; as for the second requirement, we composed the system with a few *collections*, and made them to be correlated by the same key field or ObjectID. Based on these policy, we constructed the transaction model of the inventory management shown in Fig. 10.

In the following, we show only the necessary data fields extracted from the actual data fields for the sake of simplification. In Fig. 10, “Parts inventory” *collection* (below, $Parts\ inventory$) saves each parts quantity in the parts shelf; “Parts Delivery” *collection* (below, $Parts\ delivery$) saves the delivered parts quantities, and it has the following fields {order_ID, product_ID, parts_ID, necessary_quantity, shortage_quantity, imageID}; “Image” *collection* (below, $Image$) saves {image_name, image_data} of the parts shelf. Here, imageID is the ObjectID of the $Image$. As for $Parts\ delivery$ at the planning time, {order_ID, product_ID, parts_ID, shortage_quantity} is saved, and the value of {necessary_quantity} is also set to {shortage_quantity}.

$Delivery$ is the transaction which executes the delivery processing of the parts from the parts shelf to the assembly field. And, in Fig. 10, there are transactions $Delivery(A)$ and $Delivery(B)$. They correspond to the product A and B in Fig. 5 respectively. That is, they reduce the parts quantity from $Parts\ inventory$ on the basis of the necessary amount for product A or B, and save the image and video after parts delivery into $Image$. Also, they update {shortage_quantity} of $Parts\ delivery$ according to the delivery of parts. Here, in order to process as a transaction, it performs the delivery processing by each product unit of each order. For example, in the case of product A in Fig. 5, 60 parts X and 20 parts Y is delivered. Then, if any parts is insufficient for its delivery, then no delivery is executed. That is, the $shortage\_quantity$ value is calculated by the following equation.

$$shortage\_quantity = \begin{cases} 0 & (All\ parts\ supplied) \\ at\ plan & (otherwise) \end{cases}$$

Transaction $Inventory\ count$ calculates the total quantity of parts X, Y, Z in two *collections*: $Parts\ inventory$ and $Parts\ delivery$. This process is executed as a single transaction for each parts.

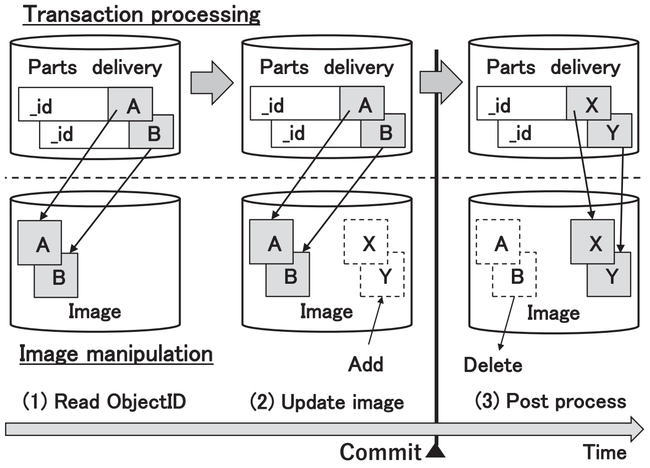


Figure 11: Access method for the enormous data

The concrete requirements to MongoDB in order to implement this model are as follows. First, the concurrency control of the transactions on two *collections* should be performed. That is, $Parts\ inventory$ and $Parts\ delivery$ are updated simultaneously, and queried. Concretely, while the transaction $Delivery$ is updating these *collections*, $Inventory\ count$ is querying these *collections* to calculate the total quantity of the parts as shown in Fig. 8. Here, both of $Delivery$ and $Inventory\ count$ should be executed as a single transaction respectively. That is, if the latter queries the anomaly state of the parts such that one *collection* has already updated and another *collection* has not updated yet, the incorrect inventory quantity is calculated.

3.4 Transaction Processing Method for Enormous Data

As for the third requirement, the images and videos are manipulated to confirm the inventory shelf. In particular, the image and video size becomes very large in the following cases where the long time video data is necessary: the status of the parts shelf shown in (a) of Fig. 7 must be confirmed from various side; various kinds of parts are delivered simultaneously from the parts shelf shown in (b). Therefore, the elapsed time of transaction to manipulate the enormous image and video becomes so long. As a results, since the lock is used in our transaction feature, the extreme latency is expected in the case where the plural transactions executed concurrently.

For this problem, we used an optimistic locking utilizing the ObjectID of $Image$ for the implementation of this method as shown in Fig. 11, in order to reduce the lock time to save the enormous data. Here, the ObjectID is the identifier of the *document* in MongoDB as shown in Section 2.1. So, it can be used instead of the time stamp, and it is used as the link that specifies the reference *document* in MongoDB. Therefore, its procedure is as follows. Firstly, at (1) in Fig. 11, {imageID} of the *document* in *Parts delivery* that is the ObjectID A and B is queried, by which the corresponding *documents* of *Image* are referred. Next, at (2), the new image or image X and Y are added to *Image*, then the transaction to update *Parts delivery* is begun. So, the target documents of *Parts delivery* are locked, and above-mentioned ObjectIDs are queried again: if these ObjectIDs have not been changed, these ObjectIDs are changed to X and Y and its commit is performed; if the ObjectIDs are changed, that is, these *documents* of $image$ have been changed by the other transactions, its rollback is performed. Then, the post process in (3) is performed: in the case of the former, the images and videos having ObjectID A and B are deleted; in the case of the latter, the images and videos having ObjectID X and Y are deleted.

By this method, above-mentioned enormous data manipulation can be separated from the lock period of the transaction, and the lock is performed only while the transaction to manipulate *Parts delivery*. As a result, the lock period can be shortened, and the manipulation of several images and videos can also be performed as a single transaction. By the way, though only the case of update is shown in Fig. 11, the case of addition and deletion of images and videos can be performed similarly: as for the former, firstly the images and videos are added, then the transaction to update *Parts delivery* is performed; as for the latter, firstly the transaction is performed, then the images and videos are deleted.

4 IMPLEMENTATIONS AND EVALUATIONS

4.1 Implementation of Inventory Management Model

We implemented the prototype of the inventory management model shown in Fig. 10 on a stand-alone PC. Its implementation environment is as follows: CPU is i7-6700 (3.41 GHz); memory is 16GB; disk is SSD memory of 512GB; OS is Windows 10. We adopted MongoDB (Ver. 3.3.6) for the database; Java (Ver. 1.8.0_73) for programming; MongoDB Java driver (Ver. 2.14.2) to access MongoDB from Java program. The above-mentioned three transaction programs are performed simultaneously using Thread class of Java, and the transaction feature shown in Section 2.1 was used for their concurrency control, which had been implemented by our previous study. And, we used GridFS class of MongoDB Java driver to store the image and video data to MongoDB [15].

Also, we implemented MongoDB update transaction by the following two methods to evaluate the deterioration of conflicts associated with saving the image and video data. The first method is shown in (2) of Fig. 12, and the image and video data is saved to MongoDB as a part of the transaction, that is, it does not use the method for the enormous data shown in Section 3.4.

Its procedure is as following. Firstly, to confirm whether there is sufficient stock in the inventory shelf, the transaction $Delivery$ query $Parts\ inventory$ query it by using the exclusive lock. So, the conflict between other transactions may occur henceforth. Then, it update {shortage_quantity} of the corresponding *document* in $Parts\ delivery$ to 0. Next, it reduces {storage_quantity} of $Parts\ inventory$, and save the image and video data to $Image$. Finally, it executes the commit. Incidentally, we excluded the case of shortage of inventory in this experiment. In addition, we set the delay between

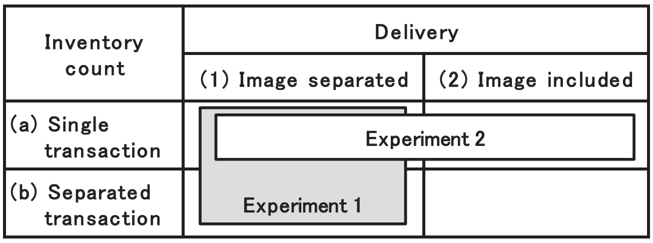


Figure 12: Program structure and experiment

the above-mentioned access of the two *collections* in order to confirm the occurrence of the conflicts. Also, in the case where the conflict occurs, the transaction performs the retry after a certain waiting time.

The second method is shown in (1) of Fig. 12 which is mentioned in Section 3.4. And, it is similar to the first method, except that it saves the image and video data before updating the *document* in $Parts\ inventory$ and $Parts\ delivery$. In other words, since the image and video data is saved prior to the start of the transaction, any *document* is not locked by this image and video data manipulation.

Then, as for the query transaction $Inventory\ count$, we also implemented it by the following two methods, in order to evaluate the difference between the execution as the single transaction and as the multiple transactions ("separated transaction" in Fig. 12 (b)). Here, the latter corresponds to the MongoDB's method such as "findOne".

In the first method, in order to prevent the *collections* to be updated by other transactions during its query, it queries each *collection* by using the shared lock. And, based on the query results, it calculates the sum of the parts. In this way, after it completes the processing, it executes the commit. Then, after waiting for a certain time, the next transaction is started to query another parts.

The second method is similar to the first method, except that it executes the commit after querying $Parts\ inventory$; then it queries $Parts\ delivery$. That is, it separates the query processes into two transactions. We show these two methods in (a) and (b) in Fig. 12 respectively.

4.2 Experiments and Evaluations

We conducted the experiments by the implementation program of the inventory management model, and evaluated the methods. The setting and procedure of the experiments are as follows. We set the sufficient inventory quantity of each parts as the initial value of $Parts\ inventory$. Also, we saved enough order data into $Parts\ delivery$.

Then, we started the three transactions in Fig. 10 at the same time: two update and one query transactions. As for the update transactions, we set 100 msec for the delay time between update the two *collections*. We also set 100 msec for the delay of next transaction start. As for the query transaction, we also set 250 msec for the delay of next transaction start. And, for both transactions, we set 50 msec for the delay before the retry when the conflict occurs. In the experiment, we executed each update transaction 12 times, and the query transaction 14 times. We used the same image data for every transaction in order to simplify the evaluation. And, its size is 3.3MB.

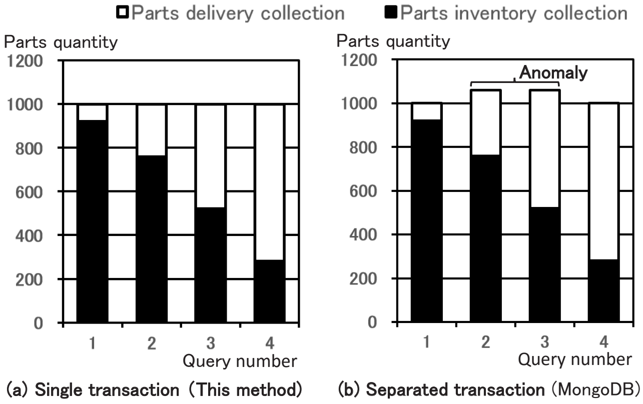


Figure 13: Result of experiment 1

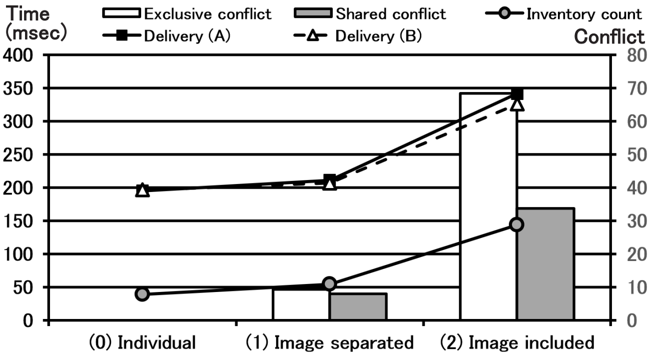
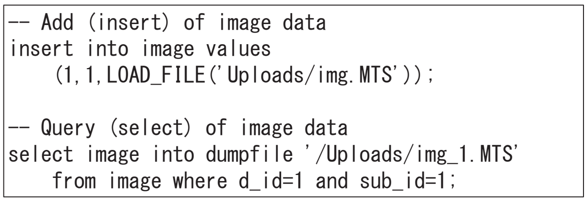


Figure 14: Result of experiment 2

For the evaluations about the first requirement, we conducted the following experiment 1 and 2. In experiment 1, we conducted the comparative evaluation between the methods (a) and (b) in (1) of Fig. 12, and Fig. 13 shows its results. Here, "Parts quantity" shows sum of the following quantities queried by *Inventory count* transaction: one was the quantity in *Parts delivery* which was as of after the delivery; the other was in *Parts inventory* which was as of before the delivery. And, Fig. 13 shows the query results at 4 check point when the quantity in *Parts inventory* of the both graph was equal. Here, the total parts quantity of the both must be always constant. And, as for (a), the query result of the total is always constant. However, as for (b), the query result is increased depending on the query timing. That is, since the query process was separated into two transactions, the anomaly occurred by querying the halfway state. As a result, it was confirmed that the isolation of the transactions is maintained also in this application field, by our transaction feature.

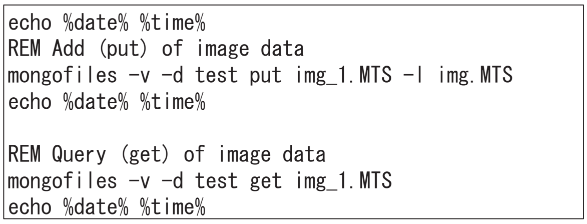
Figure 14 shows the results of experiment 2, in which we conducted the comparative evaluation between the methods (1) and (2), in (a) of Fig. 12: as for (1), the image data was saved before the transaction start as shown in Fig. 11, so the time period of transaction was shortened; as for (2), the image data was saved inside the transaction. Incidentally, this experiment was conducted in the case where image data was added. Here, this experiment was performed three times for each case, and Fig. 14 shows the average of these results.



```
-- Add (insert) of image data
insert into image values
    (1,1,LOAD_FILE('Uploads/img.MTS'));

-- Query (select) of image data
select image into dumpfile '/Uploads/img_1.MTS'
    from image where d_id=1 and sub_id=1;
```

Figure 15: Video manipulation statement of MySQL



```
echo %date% %time%
REM Add (put) of image data
mongofiles -v -d test put img_1.MTS -l img.MTS
echo %date% %time%

REM Query (get) of image data
mongofiles -v -d test get img_1.MTS
echo %date% %time%
```

Figure 16: Video manipulation command of MongoDB

Prior to this experiment, we measure the individual elapsed time of $Delivery$ and $Inventory\ count$ transaction. In this case, only one transaction is executed at the same time, and there is no conflict. We show this result in (0) of Fig. 14.

In Fig. 14, the line graph shows the change of the elapsed time for each transaction; the bar graph shows the number of the conflicts occurred for exclusive lock and shared lock respectively. (1) of Fig. 14 shows the experimental result of the method of (1) in Fig. 12. As the result in this case, the number of each conflict was about 10; the increase of elapsed time from (0) was about 10%. On the other hand, as shown in (2), in the case where the image data is stored as a part of the transaction, the number of the exclusive and shared conflict was about 70 and 30 respectively; the elapsed time of $Delivery$ transaction became 1.7 times longer than (0); $Inventory\ count$ transaction became 2.7 times longer. Therefore, as for the enormous data such as image and video data, the conflicts could be reduced by the method shown in Fig. 11.

4.3 Comparison Evaluations of Image and Video Data Access Performance

For the evaluations of access performance of image and video, we conducted the following experiment. That is, it is the performance comparison about the enormous video data between MySQL and MongoDB. Here, as shown in Section 2.2, though there are some restrictions as for MySQL, it can save the video as the binary data. Therefore, this experiment was performed using video data up to 1 GB, which is within the restriction of MySQL. We used BLOB type in MySQL, and GridFS interface of MongoDB.

The videos were shot by the digital camera, and its data size was about 121 MB per minute. In this evaluations, we used the data of 1, 2, 4 and 8 minutes. And, as for each data we measured the following elapsed time, respectively: firstly, we saved the data into the database, then queried the data. We performed these data manipulations outside of the transaction processing as shown in Fig. 11. As for MySQL, we performed the SQL statement by the MySQL monitor, and grasped their elapsed times by the displayed execution time on the monitor. We show the statements in Fig. 15: we added the data into the database by the “insert” statement, then loaded the data into the file by the “select into dumpfile” statement. Here, the folder name is simplified. And, as for MongoDB, we performed the batch file as shown in Fig. 16: we added the data into the database by the “put” command, then query the data by the “get” command; their elapsed times were grasped by displaying the system time using the “echo” command.

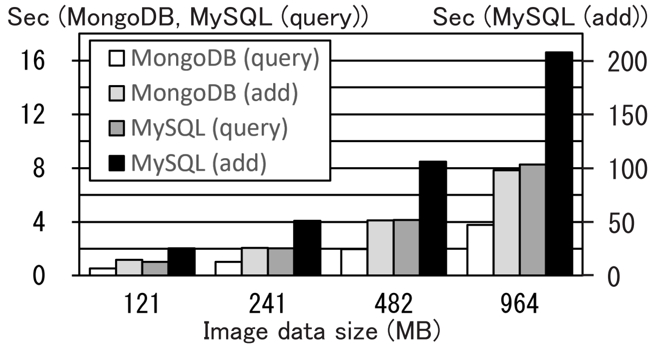


Figure 17: Evaluation result of performance comparison

Table 2: Comparison of man-hours per year

No	Classification	CapEx *	OpEx	Total
(1)	No systematization		1,200	1,200
(2)	Inventory planning	72	60	132
(3)	Image management	72	4	76

*: Man-hours per year when the life cycle is 3 years.

We show the experimental result in Fig. 17. It shows the average time of three executions as for each processing. The elapsed time increased in proportion to the amount of data in all the cases. In Fig. 17, the elapsed time of additional case in MySQL is shown by the right axis of the graph. It was about 25 times the additional case in MongoDB: in the case where the additional data size was 964 MB, the time in MySQL was more than 200 seconds; whereas the time in MongoDB was 8 seconds. In addition, the query time in MySQL was substantially the same as the additional time in MongoDB; the query time of MongoDB was about half of it.

4.4 Evaluations of Cost Performance of Proposal System

We evaluate the cost performance to introduce and operate the proposal inventory management system shown in Fig. 9. Currently, as the first step, it is planned in the target factory to gradually introduce an inventory management system for 100 common parts which are most frequently used. We show this systemization plan in Table 2, and we are advancing the systemization of inventory planning shown at (2) now. We show the detail of each systemization classification as the following.

(1) **No systematization:** This is the current situation, that is, no inventory management system has been introduced. In

this case, in order to prevent the parts shortage, it is essentially necessary to perform the stocktaking every day.

(2) **Inventory planning:** This is the systemization of the function to calculate the transition of the necessary inventory quantity due to the order information received via the EDI. That is, the theoretical inventory shown in Fig. 8 is calculated automatically. So, since it is necessary to perform the stocktaking only to grasp the difference between the actual inventory and theoretical inventory, it becomes enough to perform once per month.

(3) **Image and video management:** In addition to (2), the proposal inventory management of this study is systematized to improve the stocktaking efficiency, which utilizes the image and video of parts shelves. Incidentally, in actual operation, we are planning to take the images and videos at the time of delivery of inventory. However, we evaluate it as the monthly stocktaking for the sake of comparative evaluation with (2).

For each of these systemization, we calculated the annual man-hour from the following two perspectives: the one is capital expenditure (hereinafter, “CapEx”), which contributes to the fixed infrastructure of the company and they are depreciated over time; the other is the operational expenditures (hereinafter, “OpEx”), which do not contribute to the infrastructure itself and consequently are not subject to depreciation [29]. Here, the former is the man-hour to develop and introduce the system, and we assume that the system is depreciated in 3 years. That is, we assume the life cycle of the system is 3 years, so the man hours of CapEx shown in Table 2 are the results divided by 3. And, each man-hour of Table 2 was calculated as follows.

(a) **CapEx:** This is the man-hour to develop and introduce the system, and we calculated it based on the MRP system that has been already introduced. In (2) and (3), the man-hour for the development including the function addition and improvement after the trial use is 160 man-hours; the man-hour for introduction is mainly the data entry work, it was calculated 54 man-hours due to the ratio of the number of types of the target parts based on the introduction man-hour of the MRP system. We divided these total 214 man-hours by 3 years, which is the life cycle of the system, to calculate the annual man-hour.

(b) **OpEx:** This is the annual man-hour for the stocktaking. As for (1), we used the estimated stocktaking time of all parts obtained by the preliminary investigation. And, due to the ratio of the target parts, we calculated it as 5 man-hours for one time, then multiplied by the number of annual stocktaking regarding it as the daily work. As for (2), we calculated it as of monthly stocktaking, so it is 1/20 of (1). As for (3), we used 12 man-seconds per one type of parts, which was obtained by the experiment, and calculated in the same way as (2).

(c) **Total:** This is the total of (a) and (b).

Now, since OpEx is too large to perform as the daily regular work as shown in (1) of Table 2, it cannot be performed. On the other hand, by the systemization of the inventory planning, OpEx it is expected to be 1/20 as monthly work. So, the stocktaking is expected to be able to perform as the regular work. And, we estimate that the total man-hour including the system development and introduction can be reduced by about 90% per year. Furthermore, by adding the systemization of the image and video management (3), the man-hour of the stocktaking can be reduced, and we estimate the total man-hour can be reduced by 45% from (2).

Incidentally, the man-hour of the system development and introduction is smaller comparing to the usual commercial system. In addition, since this system can be composed on the existing PC by using only the free or existing software, the capital investment is not needed. This reason is because this system is introduced as a prototype system and we do not regard the operability such as the user interface. Instead, our students assist the system operation at the factory if necessary, and this is included the man-hour in Table 2. Incidentally, the work performed as research is not included in this man-hour, such as the method study and evaluation, technical investigation.

Then, this company evaluates the effectiveness of the system, and decides the introduction of the commercial system or using this system. For example, in the case of the preceding MRP system, they introduced a commercial system for the part related to EDI; they are using our prototype system to calculate the material cost for their estimations. The reason is because the former is related to the business connection with the other companies, and the high quality and operability are necessary; the latter needs to change the calculation flexibly according to various estimation conditions.

5 DISCUSSIONS

With the spread of the IoT, various types of enormous data are used widely. And, to store these data efficiently, GridFS interface of MongoDB is provided. So, it is expected that the enterprise system also can be more useful by using such a feature. However, as for this, there were the problems to be satisfied the following requirements as shown in Section 3.2: first, the transaction must maintain the ACID properties; second, the data manipulation must be executed without using the join operation; third, the enormous image and video manipulation must be performed as a transaction. So, there has been no its application case to the enterprise systems.

On the other hand, the inventory management work of our assisting factory was expected to be more efficiently by using the image and video data for the stocktaking. So, we conceived to apply GridFS interface of MongoDB to its production management system. And, through this application, we confirmed that the above-mentioned feature can be applied to the inventory management system. Concretely, we satisfied the requirement as following: first, we used the transaction feature which we had developed as the previous study; second, we connected *collections* by the reference using ObjectId or same key field; third, we composed the optimistic locking feature by utilizing the source *document* linked to the image and video *document*.

As a result, we confirmed MongoDB could be applied even to the enterprise system through the experiments, that is, it satisfied the above-mentioned requirements. First, as shown in Fig. 13, the anomaly of the transaction to update the plural data could be avoided by our transaction feature. Second, the collections could be composed to refer each other including the inventory data *collection* and the image and video *collection*, as shown in Fig. 11. That is, the reference using ObjectID and so on could be used instead of the join operation. Third, as shown in Fig. 14, the enormous image and video manipulation could be performed as a transaction by the optimistic locking shown in Fig. 11. In addition, its performance is better than MySQL as shown in Fig. 17.

Now, with the development of the IoT, the databases, which can handle the wide diversity of data, is expected to spread to the enterprise systems. Along with this, it is expected that many devices access the NoSQL database concurrently like this system, too. Therefore, we consider that the needs of transaction feature for the NoSQL databases would grow more. And, by this feature, we consider that the application field of the databases can be expanded to the field where the problems has occurred by using not only the RDBMS but also the conventional MongoDB.

Next, as for the efficiency by using the image and video data, we showed the case study of the stocktaking in the inventory management. As shown in OpEx of Table 2 (2) and (3), it is estimated that its man-hour can be drastically reduced. This reason is because the stocktaking work could be changed from counting the actual parts to judging that the actual inventory was sufficient by the flexible human vision.

And, there was the requirements to introduce this system into the target company, that is, the small and medium-sized company: the workload of the personnel should not increase; the system could be implemented at a low cost. As a result, as shown in Section 4.4, we could satisfy these requirement, though it was a prototype. Therefore, we consider that the system that utilizes the various type of data such as images and videos is useful for such companies. Furthermore, we consider it is useful for the other various fields.

6 CONCLUSIONS

Currently, various type of data can be used by the spread of the IoT and the development of the NoSQL database. However, to apply the NoSQL database to the enterprise systems, there were some problems such as the transaction feature. In this paper, we showed the application case of MongoDB which is a kind of NoSQL database to the production management system. We implemented the function mainly for the stocktaking as the prototype system, and we are advancing to introduce this system to the target factory. As a result, we confirmed MongoDB could be applied to the enterprise system by equipping the transaction feature maintaining the ACID properties; the function of NoSQL database such as the manipulation of the enormous data is useful even in the enterprise systems.

The feature challenge will focus on the improvement of the efficiency of actual production management system operations such as the numerical, image and video data entry at the field. In addition, we will confirm that this transaction feature can be implemented in the distributed database environment; the enormous data manipulation can be performed more efficiently in this environment.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 15K00161.

REFERENCES

[1] K. Banker, “MongoDB in Action,” Manning Pubns Co. (2011).

[2] M. Bertolini, et al., “Reducing out of stock, shrinkage and overstock through RFID in the fresh food supply chain: Evidence from an Italian retail pilot,” International Journal of RF Technologies, Vol. 4, No. 2, pp. 107–125 (2013).

[3] R. Cattell, “Scalable SQL and NoSQL data stores,” ACM SIGMOD Record, Vol. 39, No. 4, pp. 12–27 (2011).

[4] I.J. Chen, “Planning for ERP systems: analysis and future trend,” Business process management journal, Vol. 7, No. 5, pp. 374–386 (2001).

[5] M. Chen, S. Mao, and Y. Liu, “Big data: A survey,” Mobile Networks and Applications, Vol. 19, No. 2, pp. 171–209 (2014).

[6] A. Eisenberg, and J. Melton, “SQL:1999, formerly known as SQL3,” ACM Sigmod record, Vol. 28, Issue 1, pp. 131–138 (1999).

[7] J. Gray, and A. Reuter, “Transaction Processing: Concept and Techniques,” San Francisco: Morgan Kaufmann (1992).

[8] S. Hiremath, G. Yang, and K. Mankodiya, “Wearable Internet of Things: Concept, architectural components and promises for person-centered healthcare,” EAI 4th International Conference on Wireless Mobile Communication and Healthcare (2014).

[9] C.L. Iacovou, I. Benbasat, and A.S. Dexter, “Electronic data interchange and small organizations: adoption and impact of technology,” MIS quarterly, Vol. 19, No. 4, pp. 465–485 (1995).

[10] T. Kudo, M. Ishino, K. Saotome, and N. Kataoka, “A Proposal of Transaction Processing Method for MongoDB,” Procedia Computer Science, Vol 96, pp. 801–810 (2016).

[11] D. Laney, “3D Data Management: Controlling Data Volume, Velocity and Variety,” META Group (2012), http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf (reffered May 5, 2017).

[12] H. Garcia-Molina, and K. Salem, “SAGAS,” Proc. of the 1987 ACM SIGMOD Int. Conf. on Management of data, pp. 249–259 (1987).

[13] K. Hong-yan, “Design and realization of internet of things based on embedded system used in intelligent campus,” International Journal of Advancements in

Computing Technology, Vol. 3, No. 11, pp. 291–298 (2011).

[14] MongoDB Inc., “The MongoDB 3.4 Manual,” http://docs.mongodb.org/manual/ (reffered May 5, 2017).

[15] MongoDB Inc., “MongoDB API Documentation for Java,” http://api.mongodb.org/java/ (reffered May 5, 2017).

[16] MongoDB Inc., “Write Operation Performance,” https://docs.mongodb.com/v3.4/core/write-performance/ (reffered May 5, 2017).

[17] Oracle Corp., “MySQL 5.7 Reference Manual,” http://dev.mysql.com/doc/refman/5.7/en/ (reffered May 5, 2017).

[18] J. Pokorny, “NoSQL databases: a step to database scalability in web environment,” International Journal of Web Information Systems, Vol. 9, No. 1, pp. 69–82 (2013).

[19] D.R. Rebecca, and I. E. Shanthi, “A NoSQL Solution to efficient storage and retrieval of Medical Images,” International Journal of Scientific & Engineering Research, Vol. 7, No. 2, pp. 545–549 (2016).

[20] E. Redmond, and J.R. Wilson, “Seven Databases in Seven Weeks: A guide to Modern Databases and the NoSQL Movement,” Pragmatic Bookshelf (2012).

[21] R. Sears, C. Van Ingen, and J. Gray, “To blob or not to blob: Large object storage in a database or a filesystem?,” arXiv preprint cs/0701168 (2007).

[22] K. Seguin, “The Little MongoDB Book” (2011), http://openmymind.net/mongodb.pdf (reffered May 5, 2017).

[23] L.A.B. Silva, et al., “Medical imaging archiving: A comparison between several NoSQL solutions,” Int. Conf. on Biomedical and Health Informatics, pp. 65–68 (2014).

[24] The Small and Medium Enterprise Agency, “2015 White Paper on Small and Medium Enterprises in Japan and White Paper on Small Enterprises in Japan (outline),” (2015), http://www.chusho.meti.go.jp/pamflet/hakusyo/H27/download/2015hakushogaiyou_eng.pdf (reffered May 5, 2017).

[25] S.S. Sriparasa, “JavaScript and JESON Essentials,” Packt Pub. Ltd. (2013).

[26] M.P. Stević, B. Milosavljević, and B. R. Perišić, “Enhancing the management of unstructured data in e-learning systems using MongoDB,” Program, Vol. 49, No. 1, pp. 91–114 (2015).

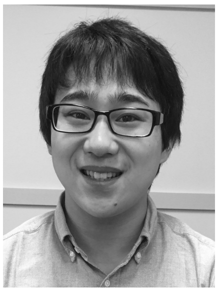
[27] M. Stonebraker, and C. Ugur, “One size fits all”: an idea whose time has come and gone,” Proc. of 21st Int. Conf. on Data Engineering, pp. 2–11 (2005).

[28] M. Stonebraker, et al., “The end of an architectural era:(it’s time for a complete rewrite),” Proc. of 33rd Int. conf. on Very large data bases. VLDB Endowment, pp. 1150-1160 (2007).

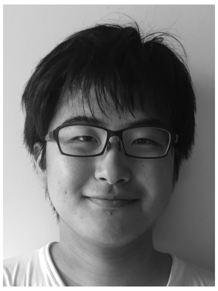
[29] S. Verbrugge, et al., “Modeling operational expenditures for telecom operators,” Proc. of Conf. on Optical Network Design and Modeling, pp. 455–466 (2005).

(Received October 9, 2016)
(Revised May 9, 2017)

Tsukasa Kudo received the M.E. from Hokkaido University in 1980 and the Dr.Eng. in industrial science and engineering from Shizuoka University, Japan in 2008. In 1980, he joined Mitsubishi Electric Corp. He was a researcher of parallel computer architecture, an engineer of application packaged software and business information systems. Since 2010, he is a professor of Shizuoka Institute of Science and Technology. Now, his research interests include database application and software engineering. He is a member of IEIEC and Information Processing Society of Japan.



Yuki Ito is currently working toward a B.I. degree at Shizuoka Institute of Science and Technology. His research interests include IoT, web system and SEO.



Yuki Serizawa is currently working toward a B.I. degree at Shizuoka Institute of Science and Technology. His research interests include database application and production management system.

Best-Time Estimation for Regions and Tourist Spots using Phenological Observations with Geotagged Tweets

Masaki Endo[*,**], Yoshiyuki Shoji[***], Masaharu Hirota[****], Shigeyoshi Ohno[*], and Hiroshi Ishikawa[**]

[*]Division of Core Manufacturing, Polytechnic University, Japan
{endou, ohno}@uitec.ac.jp
[**]Graduate School of System Design, Tokyo Metropolitan University, Japan
ishikawa-hiroshi@tmu.ac.jp
[***] Program-Specific Researcher, Academic Center for Computing and Media Studies, Kyoto University, Japan
shoji.yoshiyuki.5m@kyoto-u.ac.jp
[****] Department of Information Engineering, National Institute of Technology, Oita College, Japan
m-hirota@oita-ct.ac.jp

Abstract - In recent years, social network services (SNS) such as Twitter have become widely used, attracting great attention for many reasons. An important characteristic of Twitter is its real-time property. Twitter users post huge volumes of Twitter posts (*tweets*) related to daily events in real time. We assume that the tweet contents depend on the region, season, and time of day. Therefore, the possibility exists of obtaining valuable information for tourists from tweets posted during travel. As described in this paper, we propose a method to estimate regional best times for viewing flower blossoms from tweets including flower names. Our proposed method analyzes the number of tweets using a moving average. Additionally, we particularly examine geotagged tweets. Our experiments compare the best-time viewing estimated using our method to the flowering date and the full bloom date of cherry blossoms that the Japan Meteorological Agency has observed and posted. We conducted an experiment using data for the best-time viewing cherry blossoms during 2015 and 2016. Results confirmed that the proposed method can estimate the full bloom period accurately.

Keywords: trend estimation; phenological observation; Twitter

1 INTRODUCTION

In recent years, because of rapid performance improvement and the dissemination of various devices such as smart phones and tablets, diverse and vast data are generated on the web. Particularly, social networking services (SNS) have become prevalent because users can post data and various messages easily. According to the 2014 Communications Usage Trend Survey of the Ministry of Internal Affairs and Communications (MIC) [1], the percentage of Japanese people aged 13–39 years old using SNS is greater than 60%, the figure for people 40–49 years is higher than 50%. Twitter [2], an SNS that provides a micro-blogging service, is used as a real-time communication tool. Numerous tweets have been posted daily by vast numbers of users. Twitter is therefore a useful medium to obtain, from a large amount of information posted by many users, real-time information corresponding to the real world.

Here, we describe the provision of information to tourists using the web. Before SNS were used, local governments, tourism organizations, and travel companies provided regional tourism information using web pages. After SNS became widely used, they also undertook efforts to disseminate more detailed information related to respective tourist spots. The information is useful for tourists, but providing timely and topical travel information entails high costs for the information provider because they must update the information continually. Today, providing reliable information related to local travel is not only strongly demanded by tourists, but also local governments, tourism organizations, and travel companies, which bear high costs of providing the information.

Tourists also want real-time information and local unique seasonal information posted on web sites, according to a survey study of IT tourism and services to attract customers [3] by the Ministry of Economy, Trade and Industry (METI). Current web sites provide similar information in the form of guide books. Nevertheless, the information update frequency is low. Because local governments, tourism associations, and travel companies provide information about travel destination local unit independently, it is difficult for tourists to collect information for “now” tourist spots.

Therefore, providing current, useful, real-world information for travelers by capturing the change of information in accordance with the season and time zone of the tourism region is important for the travel industry. As described herein, we define "now" as information for tourism and disaster prevention required by travelers during travel, such as a best flower-viewing time and festivals and local heavy rains.

We propose a method to estimate the best time for phenological observations for tourism such as the best-time viewing cherry blossoms and autumn leaves in each region by particularly addressing phenological observations assumed for “now” in the real world. Tourist information for the best time requires a peak period, which means that the best time is not a period after and before falling flowers, but

ISSN1883-4566 ©2017 - Informatics Society and the authors. All rights reserved.

a period to view blooming flowers. Furthermore, the best times differ among regions and locations. Therefore, it is necessary to estimate a best time of phenological observation for each region and location. Estimating the best-time viewing requires the collection of much information having real-time properties. For this study, we use Twitter data obtained for many users throughout Japan.

The remainder of the paper is organized as follows. Chapter 2 presents earlier research related to this topic. Chapter 3 describes our proposed method for estimation of best time of phenological observations. Chapter 4 describes experimentally obtained results for our proposed method and a discussion of the results. Chapter 5 summarizes the contributions and future work.

2 RELATED WORK

The amount of digital data is expected to increase greatly in the future because of the spread of SNS. Reports describing studies of the effective use of these large amounts of digital data are numerous. Some studies have used microblogs to conduct real-world understanding and prediction by analyzing information transmitted from microblogs. Kleinberg [5] detected a "burst" of keywords signaling a rapid increase in time-series data. Ochiai et al. [6] proposed a disambiguation method for family names that are also used as place names using dynamic characteristic words of topics that vary from period to period, including static characteristic words and locations that are independent of specific seasonal variation according to the location as a target of microblog. Kurata et al. [7] developed a system to detect events in real space using geotagged tweets. This system can grasp what events occur in time and place by the top 10 of frequent word extraction conducted in each time zone. Sakaki et al. [8] proposed a method to detect events such as earthquakes and typhoons based on a study estimating real-time events from Twitter. Kunneman et al. [9] proposed a method to differentiate among tweets posted before, during, and after a soccer match using machine learning. Hurriyetoglu et al. [10] proposed a method to estimate the time to a future soccer match using tweet streams with local regression over a word time series. Tops et al. [11] described a method to classify the time to an event in automatically discretized categories using support vector machines. Consequently, various methods for extracting event and location information have been discussed. Nevertheless, a method used to estimate the start and end of the full bloom period of phenological observations using tweets is controversial.

3 OUR PROPOSED METHOD

This chapter presents a description of a method of analysis for target data collection and our best-time estimation to get a guide for phenological change from Twitter in Japan. Our proposal is portrayed in Fig. 1.

3.1 Data Collection

This section presents a description of the Method of (1) data collection shown in Fig. 1. Geotagged tweets sent from Twitter are a collection target. Range geo-tagged tweets include the Japanese archipelago (120.0 ≤ longitude ≤ 154.0 and 20.0 ≤ latitude ≤ 47.0) as the collection target. Collection of these data was done using Streaming API [12], one API provided by Twitter, Inc.

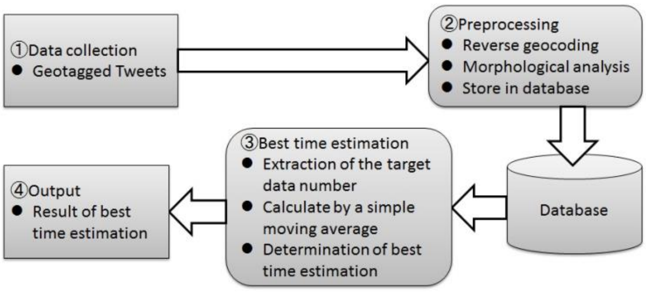
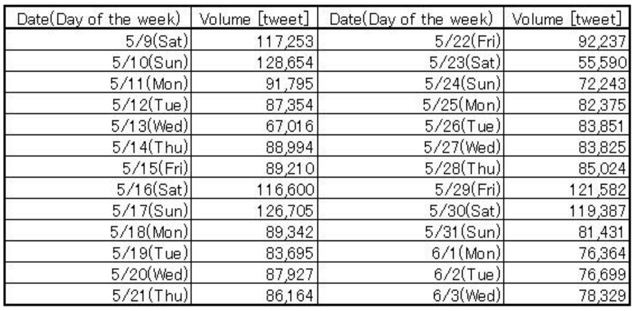

①Data collection
● Geotagged Tweets
②Preprocessing
● Reverse geocoding
● Morphological analysis
● Store in database
Database
③Best time estimation
● Extraction of the target data number
● Calculate by a simple moving average
● Determination of best time estimation
④Output
● Result of best time estimation

Figure 1: Our proposal summary

Table 1: Transition example of geotagged tweets (2015/5/9-6/3)



Date(Day of the week)	Volume [tweet]	Date(Day of the week)	Volume [tweet]
5/9(Sat)	117,253	5/22(Fri)	92,237
5/10(Sun)	128,654	5/23(Sat)	55,590
5/11(Mon)	91,795	5/24(Sun)	72,243
5/12(Tue)	87,354	5/25(Mon)	82,375
5/13(Wed)	67,016	5/26(Tue)	83,851
5/14(Thu)	88,994	5/27(Wed)	83,825
5/15(Fri)	89,210	5/28(Thu)	85,024
5/16(Sat)	116,600	5/29(Fri)	121,582
5/17(Sun)	126,705	5/30(Sat)	119,387
5/18(Mon)	89,342	5/31(Sun)	81,431
5/19(Tue)	83,695	6/1(Mon)	76,364
5/20(Wed)	87,927	6/2(Tue)	76,699
5/21(Thu)	86,164	6/3(Wed)	78,329

Next, we describe the collected number of data. The percentage of geotagged tweets among tweets originated in Japan, according to the study of Hashimoto et al. [13] as a whole is about 0.18%. Such tweets are very few among all data. However, the collected geo-tagged tweets, shown as an example in Table 1, number about 70,000, even on weekdays. On weekends there are also days on which more than 100,000 such messages are posted. We use about 30 million geo-tagged tweets from 2015/2/17 through 2016/6/30. For each day of collection, the number during the period covered was about 72,000. We calculated the best time for flower viewing, as estimated by processing the following sections using these data.

3.2 Preprocessing

This section presents a description of the method of (2) preprocessing shown in Fig. 1. Preprocessing includes reverse geocoding and morphological analysis, as well as database storage for data collected through the processing described in Section 3.1.

Reverse geocoding identified prefectures and municipalities by town name from latitude and longitude information of the individually collected tweets. We use a simple reverse geocoding service [14] available from the National Agriculture and Food Research Organization in this process: e.g., (latitude, longitude) = (35.7384446, 139.460910) by reverse geocoding becomes (Tokyo, Kodaira City, Ogawanishi-cho 2-chome).

Morphological analysis divides the collected geo-tagged tweet morphemes. We use the “Mecab” morphological analyzer [15]. By way of example, "桜は美しいです” (in English “Cherry blossoms are beautiful.”)" is divided into "(桜 / noun), (は / particle), (美しい / adjective), (です / auxiliary verb), (。 / symbol)".

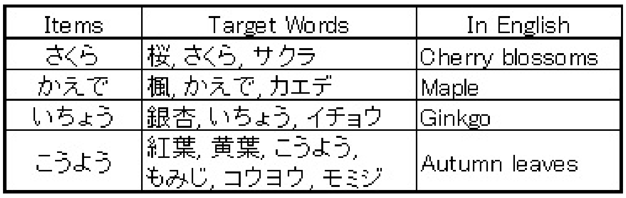
Preprocessing performs the necessary data storage for the best-time viewing, as estimated Based on results of the processing of the data collection and reverse geocoding and morphological analysis. Data used for this study were the tweet ID, tweet post time, tweet text, morphological analysis result, latitude, and longitude.

3.3 Estimating the Best-Time Viewing

This section presents a description of the method of (3) best-time estimation presented in Fig. 1. Our method for estimating the best-time viewing processes the target number of extracted data and calculates a simple moving average, yielding an inference of the best flower-viewing time. The method defines a word related to the best-time viewing, estimated as the target word. The target word can include Chinese characters, hiragana, and katakana, which represents an organism name and seasonal change, as shown in Table 2.

Next, we describe a simple moving average calculation, which uses a moving average of the standard of the best-time viewing judgment. It calculates a simple moving average using data aggregated on a daily basis by the target number of data extraction described above. Figure 2 presents an overview of the simple moving average of the number of days.

Table 2: Examples of the target word



Items	Target Words	In English
さくら	桜, さくら, サクラ	Cherry blossoms
かえで	楓, かえで, カエデ	Maple
いちょう	銀杏, いちょう, イチョウ	Ginkgo
こうよう	紅葉, 黄葉, こうよう, もみじ, コウヨウ, モミジ	Autumn leaves

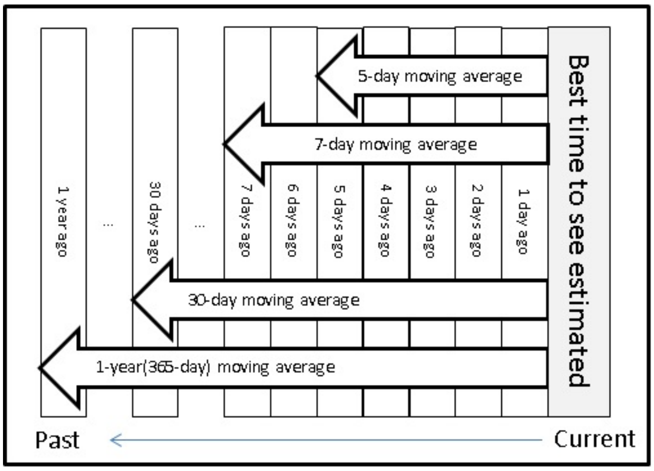


Figure 2: Number of days simple moving average

We calculate the simple moving average in formula (1) using the number of data going back to the past from the day before the estimated date of the best-time viewing.

$$X(Y) = \frac{P_1+P_2+\cdots+P_Y}{Y} \tag{1}$$

$X(Y)$: Y $d$ay moving average
P_n: Number of data of n days ago
Y: Calculation target period

The standard lengths of time we used for the simple moving average were a 7-day moving average and 1-year moving average. As shown in Table 1, since geo tag tweets tend to be more frequent at weekends than on weekdays, a moving average of 7 days is taken as one of estimation criteria. And the phenomenological observation is based on the one-year moving average as the estimation criterion because there are many "viewing" events every year, such as "viewing of cherry blossoms", "viewing of autumn leaves" and "harvesting month".

In addition to the 7-day moving average and the 1-year moving average, we also explain the moving average of the number of days specified for each phenological. In this study, we set the number of days of moving average from specified biological period of phenological.

As an example, we describe cherry blossoms. The Japan Meteorological Agency (JMA) [16] carries out phenological observations of "Sakura," which yields two output items of the flowering date and the full bloom date observation target. "Sakura of flowering date" [17] is the first day of blooming 5–6 or more wheels of flowers of a specimen tree. "Sakura in full bloom date" is the first day of a state in which about 80% or more of the buds are open in the specimen tree. In addition, "Sakura" is the number of days from general flowering until full bloom: about 5 days. Therefore, "Sakura" in this study uses a 5-day moving average, which is standard.

Next, we describe an estimated judgment of the best-time viewing, which was calculated using the simple moving average (7-day moving average, 1-year moving average, and another biological moving average). It specifies the two conditions as a condition of an estimated decision for the best-time viewing.

Condition 1 uses the 1-year moving average and the number of tweets containing the organism name of each day. Compare the number of tweets on each day and the 1-year moving average calculated for each day as shown in Equation 2. The day when the number of tweets on each day exceeds the 1-year moving average is the day when the condition1 holds.

$$P_1 \geqq X(365) \tag{2}$$

For condition 2, we use equation 3 to make a judgment using 7-day moving average and biological moving average. Here, A in Equation 3 refers to the shorter number of days by comparing "moving average of 7 days" and "moving average of bioequivalence". B is a long number of days.

$$X(A) \geqq X(B) \tag{3}$$

As an example, cherry blossoms use a 7-day moving average and 5-day moving average, so A is 5 days and B is 7 days. This determines the date on which the moving average of a short number of days exceeds the moving average of a long number of days. Subsequently, it is assumed that Condition 2 is satisfied when the moving average of a short number of days exceeds the moving average of a long number of days continuously. The number of consecutive days was made equal to or more than half of the moving average of a short number of days. In the case of cherry blossoms, the 5-day moving average is shorter than the 7-day moving average. Therefore, 5 days / 2 = 2.5 days ≒ 3 days as a standard. If the 5-day moving average exceeds the 7-day moving average by 3 days or more, it shall be the date satisfying Condition 2.

Finally, we estimate the day that both Condition 1 and Condition 2 are satisfied as best time to see.

3.4 Output

This section presents a description of the method of (4) output presented in Fig. 1. Output can be visualized using a result of the best-time viewing, as estimated by processing explained in the previous section. This paper presents a visualization that reflects the best-time viewing inference results in a time-series graph. The graph shows the number of data and the date, respectively, on the vertical axis to the horizontal axis. We are striving to develop useful visualization techniques for travelers.

4 EXPERIMENTS

In this chapter, the experimental explanation for guessing the optimum time to see the flower in the proposed method described in Chapter 3 is shown. This shows the dataset used for the optimum time reasoning to see flowers that bloom completely in section 4.1. Section 4.2 shows the target word and target area used in the experiment. Section 4.3 shows experimental results on 2015 cherry blossoms viewing and 2016 cherry blossoms viewing. Section 4.4 shows the result of comparing the estimation result with observation data. Section 4.5 shows the experiment results using the sightseeing spots co-occurring in the target word.

4.1 Dataset

Datasets used for this experiment were collected using streaming API, as described for data collection in Section 3.1. Data are geo-tagged tweets from Japan during 2015/2/17 – 2016/6/30. The data include about 30 million items. We are using these datasets for experiments to infer the best time for cherry blossom viewing in 2015 and in 2016.

4.2 Estimation Experiment for Best-Time Viewing Cherry Blossoms

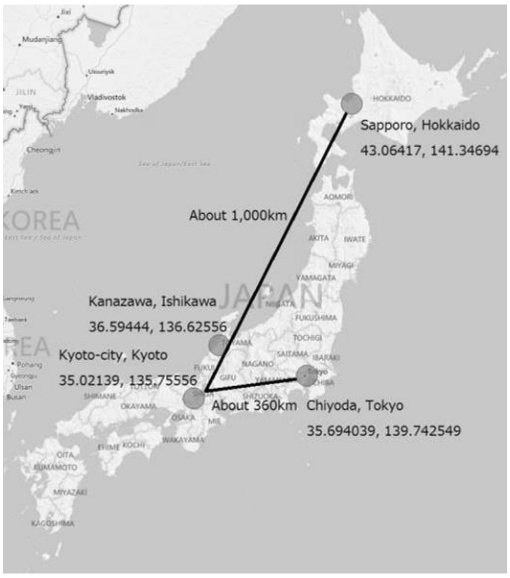


Figure 3: Position of target area

The estimation experiment to ascertain the best-time viewing cherry blossoms uses the target word in Table 2: "Sakura". The target word is “cherry blossom,” which is “桜” and “さくら” and “サクラ” in Japanese. The experimental target areas were "Tokyo," "Ishikawa", "Kyoto," and “Hokkaido." For each area, a specimen tree is used for observations by the JMA. The cities of “Chiyoda," "Kanazawa," "Kyoto," and "Sapporo" are target areas. In addition, an experiment using co-occurrence words was conducted using tweets with the item "Sakura" for many tourist spot named in Table 2, which also shows with the number of occurrences in respective regions. As described in this paper, we specifically examined "Shinjuku Gyoen", "Rikugien Garden," "Goryokaku," and "Kenroku-en".

Figure 3 presents the target area location. Kyoto and Hokkaido are separated by about 1,000 km straight line distance. Kyoto and Tokyo are about 360 km apart. Because of their latitudes, cherry trees flower later in the north in Hokkaido than in Kyoto. Moreover, higher altitudes and consequently cooler temperatures delay flowering even when locations have similar latitudes. Although issues related to altitude were not particularly addressed in this study, they are not expected to affect important results for single sites.

4.3 Target Word Results in Target Areas

Figure 4 presents experimentally obtained results for the estimated best-time viewing in 2015 using the target word cherry blossoms in the target area of "Tokyo." The dark gray bar in the figure represents the number of tweets. The light gray part represents the best-time viewing as determined using the proposed method. In addition, the solid line shows a 5-day moving average. In addition, the solid line shows a 5-day moving average. The dashed line shows a 7-day moving average. The dotted line shows the 1-year moving average. Figure 5 shows the estimated best-time viewing, as inferred from experimentally obtained results in 2016 using the target word cherry blossoms in the target area of "Tokyo."

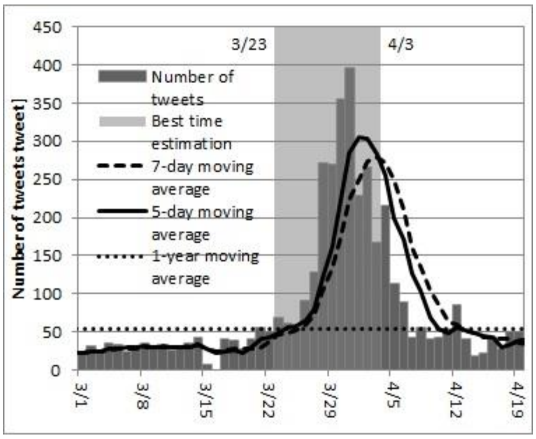


Figure 4: Results of the best time to see, as estimated by Tokyo (2015)

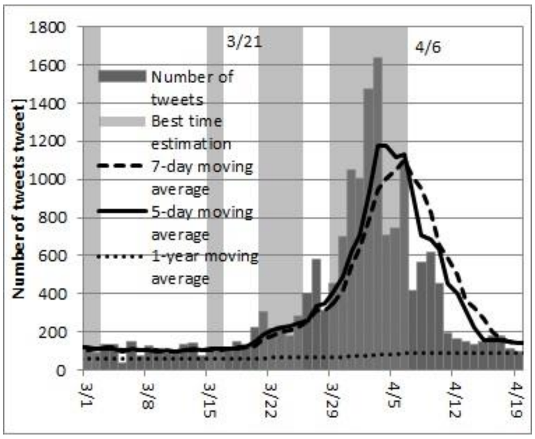


Figure 5: Results of the best time to see, as estimated by Tokyo (2016)

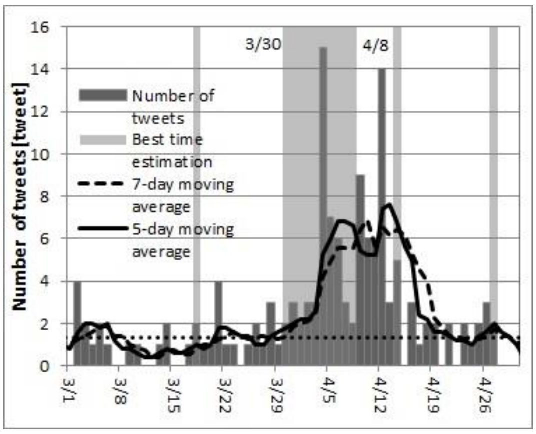


Figure 6: Results of best time to see, as estimated by Ishikawa (2015)

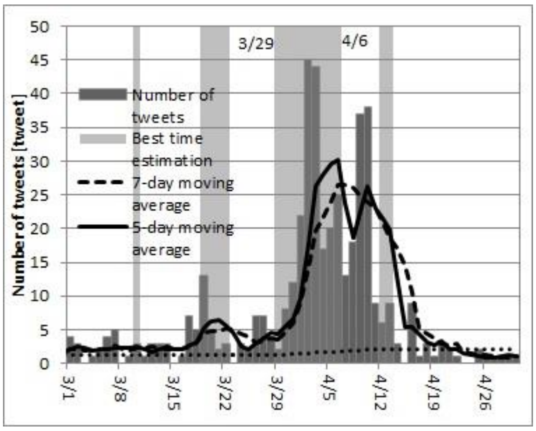


Figure 7: Results of best time to see, as estimated by Ishikawa (2016)

For Tokyo in 2015, as portrayed in Fig. 4, we obtained the greatest number of data. The greatest number of tweets per day reached about 400. Our proposed method indicates the best-time viewing as 3/23 – 4/3. Condition 1 shown in 3.3 is the day when a dark gray bar exceeds the dotted line. Condition 2 is the day when the solid line exceeds the broken line for more than 3 days. Therefore, in our proposed method, we estimated 3/23 – 4/3 which satisfy both condition 1 and condition 2 as best-time viewing.

1Our proposed method shows the best-time viewing as 3/21 – 4/6 in Fig. 5. The estimation for the best-time viewing in 2016 indicates a longer period than that in 2015, which is consistent with the trend of 2016, with low-temperature days after flowering. Tokyo of 2016, as presented in Fig. 5, also has the largest number of data in the area of the experimental subjects of 2016. More than 1,600 tweets were sent on some days, which is about four times that of 2015. Therefore, the 1-year moving average value for the rapid increase in the number of tweets is reduced. For that reason, much noise is included in the estimate of the best-time viewing.

Figure 6 presents results of 2015 for Ishikawa Prefecture. Results of 2016 for Ishikawa Prefecture are portrayed in Fig. 7. The greatest numbers of data were, respectively, 15 tweets and 45 tweets. Ishikawa data are far fewer than those of Tokyo. However, 2015 has been the best-time viewing was estimated as 3/30 – 4/8. In 2016, the best-time viewing was estimated as 3/29 – 4/6. Noise contents that are unrelated to "Sakura" organisms are also included in tweets. However, before the peak period, tweets abound for budding and flowering cherry. After the peak period, tweets related to cherry blossom leaves are prominent.

From the above, it is possible to estimate the best time by using the proposed method only using the number of occurrences of the target word. This is because the cherry trees of the target word shown in this experiment are the most frequent tweets pointing to cherry trees of creatures in Japan. In the experiment, the accuracy in the area with many tweets was relatively high. However, in addition to tweets as cherry trees of living things, it is considered that there are also tweets including cherry blossoms that become noise used for different purposes such as person's name and food. Also, in the case of another target word, there are cases where the best time to estimate the best time other than the creature is erroneously estimated. Although this paper does not mention analysis of tweet contents, it is a future subject. Also, in the case of other target words, there are cases where the best time other than the organism is erroneously estimated. Although this paper does not mention analysis of the contents of tweets, it is a future task.

4.4 Comparing Best Times for Viewing Estimation and Observed Data

Table 3 presents results of comparison between the estimated best-time viewing and JMA observation data in target area in 2015. Dates in the table are the target dates for the estimated best-time viewing. The thin gray portion of each region is the day determined as the best-time viewing: as an example, Tokyo's best-time viewing in 2015 was 3/23 – 4/3. This result represents the same day and the estimated best-time viewing thin gray part of the previous section in Fig. 4. Furthermore, the arrow indicates the period of up to "cherry blossoms in full bloom date" from "cherry flowering date" that the JMA has observed in each region. As an example, Tokyo observations are based on the specimen tree in Chiyoda. The "Sakura flowering date" of 2015 is 3/23. The "Sakura in full bloom date" of 2015 is 3/29. Specimen trees of the JMA of the experimental target area are the following. Ishikawa is Kanazawa. Kyoto is Kyoto City. Hokkaido is Sapporo. Recall and precision using the observed data and the estimated best-time viewing results are calculated for each target area for 2015 from 3/1 – 6/30 using formula (4) and formula (5).

$$Precision = \frac{Number\ of\ days\ to\ match\ the\ observed\ data}{Number\ of\ days\ in\ best\ time\ to\ see\ estimated} \quad (4)$$

$$Recall = \frac{Number\ of\ days\ to\ match\ the\ observed\ data}{Number\ of\ days\ of\ observation\ data} \quad (5)$$

The precision ratio average in Table 3 is about 20%. A low precision ratio does not include the period from full bloom to abscission. The best-time viewing is estimated as 3/30 – 4/3 for Tokyo as determined by the JMA as the best-time viewing after full blooming of cherry trees. Therefore, the result presents the possibility of providing the best-time viewing information that is necessary to complement tourist observation data of the JMA using the proposed method. However, the data are few for areas such as Kanazawa. Therefore, the moving average used to estimate the best time for flower viewing is vulnerable to extreme changes.

Moreover, Hokkaido, Tokyo, Ishikawa, and Kyoto recall is higher than that of municipal districts. These experiments use aggregate data of each whole area against observation data of a sample tree of the JMA. Chiyoda and Kanazawa are regions within prefectures. They therefore have a low recall rate because the data are fewer. Kyoto and Sapporo show no decrease of recall because many data in the region are city data. Results of this best-time estimation should be provided as tourist information in each region for which there is limited information of target areas.

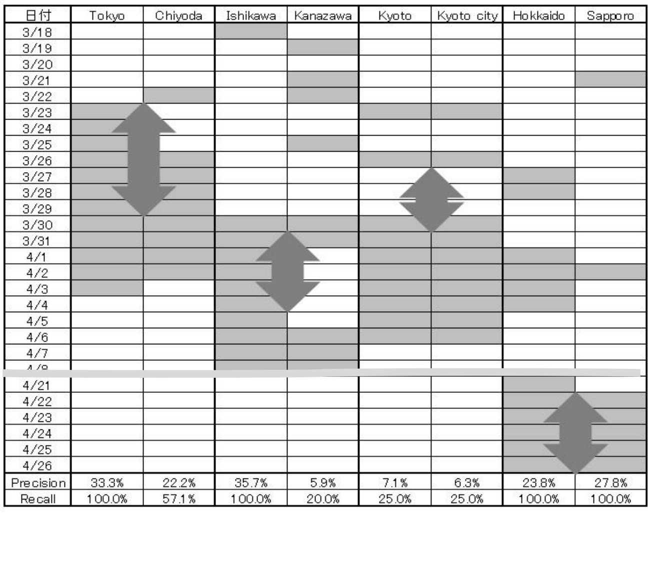
Table 4 presents experimentally obtained results for 2016. The notation is the same as that used in Table 3. The experimental period in 2016 was 3/1 – 6/30. Data of 2016 obtained using our proposed method were also confirmed best-time estimation for each region. Data confirmed the best-time estimation after full bloom observation by the JMA. Compared to 2015, 2016 was confirmed to have a long best-time duration because of low temperatures after flowering. However, precision and recall for some data loss are lower than in 2015.

From the above, the proposed method is useful for estimating the optimum time for viewing cherry blossoms in areas where about 10 tweets per day were obtained. However, since the area under study in this experiment is the capital of the prefecture, there are also relatively many tweet data. There is also a specimen tree used for JMA observation. However, in many other areas there may be regions where there are few data. Therefore, we need to verify further in other areas.

4.5 Results of the Full Bloom Estimation using the Co-occurrence Word

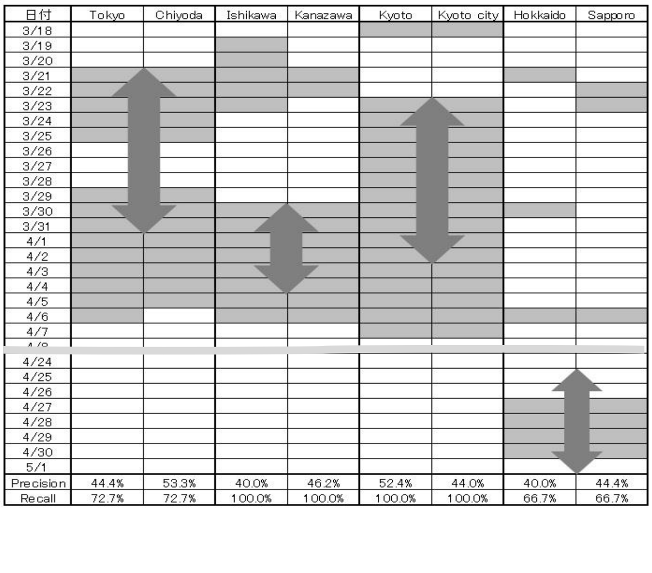
Table 5 presents estimation results for the best-time viewing in 2015 with tweets that include the tourist spot name co-occurring with the target word "Sakura". The co-occurrence words are tourist spot names used for estimation with the proposed method: "Shinjuku Gyoen," "Rikugien Garden," "Goryokaku," and "Kenroku-en". The numerical values in the table are the numbers of tweets including the target word and the co-occurrence word. The light gray part shows a date for which full bloom estimation was made

Table 3: Comparison result in target areas of the best time to see the estimated and the observed data (2015)



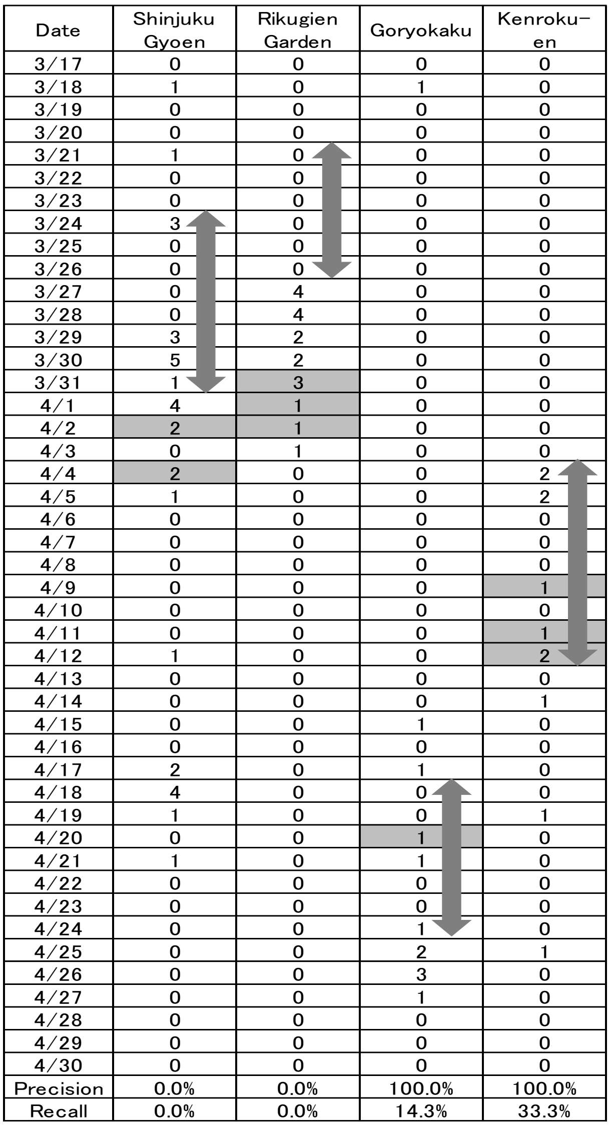
日付	Tokyo	Chiyoda	Ishikawa	Kanazawa	Kyoto	Kyoto city	Hokkaido	Sapporo
3/18			▨					
3/19				▨				
3/20								
3/21				▨				▨
3/22		▨		▨				
3/23	▨				▨	▨		
3/24	▨							
3/25	▨			▨				
3/26	▨	▨			▨	▨		
3/27	▨	▨					▨	
3/28	▨	▨					▨	
3/29	▨	▨						
3/30	▨	▨	▨	▨	▨	▨		
3/31	▨	▨	▨	▨	▨	▨		
4/1	▨	▨	▨		▨	▨	▨	
4/2	▨	▨	▨		▨	▨	▨	▨
4/3	▨		▨		▨	▨	▨	
4/4			▨		▨	▨	▨	
4/5			▨		▨	▨		
4/6			▨	▨	▨	▨		
4/7			▨	▨				
4/8			▨	▨				
4/21							▨	
4/22							▨	▨
4/23							▨	▨
4/24							▨	▨
4/25							▨	▨
4/26							▨	▨
Precision	33.3%	22.2%	35.7%	5.9%	7.1%	6.3%	23.8%	27.8%
Recall	100.0%	57.1%	100.0%	20.0%	25.0%	25.0%	100.0%	100.0%

Table 4: Comparison result in target areas of the best time to see the estimated and the observed data (2016)



日付	Tokyo	Chiyoda	Ishikawa	Kanazawa	Kyoto	Kyoto city	Hokkaido	Sapporo
3/18					▨	▨		
3/19			▨					
3/20			▨					
3/21	▨	▨	▨	▨			▨	
3/22	▨	▨	▨	▨				▨
3/23	▨	▨	▨		▨	▨		▨
3/24	▨	▨			▨	▨		
3/25	▨	▨			▨	▨		
3/26					▨	▨		
3/27					▨	▨		
3/28					▨	▨		
3/29	▨	▨			▨	▨		
3/30	▨	▨	▨	▨	▨	▨	▨	
3/31	▨	▨	▨	▨	▨	▨		
4/1	▨	▨	▨	▨	▨	▨		
4/2	▨	▨	▨	▨	▨	▨		
4/3	▨	▨	▨	▨	▨	▨		
4/4	▨	▨	▨	▨	▨	▨		
4/5	▨	▨	▨	▨	▨	▨		
4/6	▨		▨	▨	▨	▨	▨	▨
4/7					▨	▨		
4/8								
4/24								
4/25								
4/26								
4/27							▨	▨
4/28							▨	▨
4/29							▨	▨
4/30							▨	▨
5/1								
Precision	44.4%	53.3%	40.0%	46.2%	52.4%	44.0%	40.0%	44.4%
Recall	72.7%	72.7%	100.0%	100.0%	100.0%	100.0%	66.7%	66.7%

Table 5: Best time to see the estimated and tweet the number of tourist spot name to co-occurrence (2015)



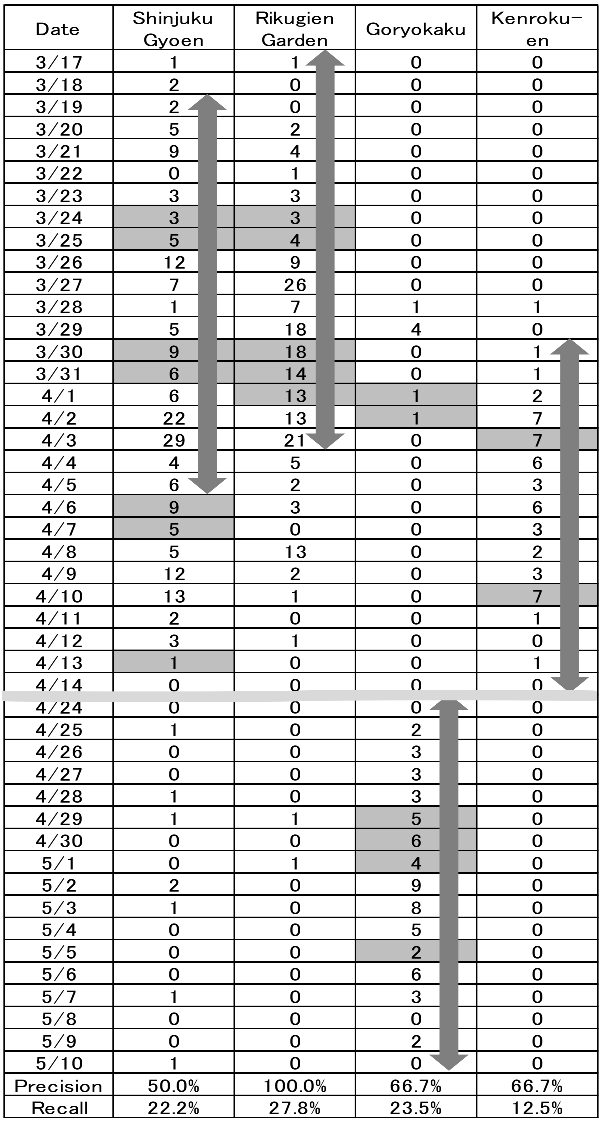
Date	Shinjuku Gyoen	Rikugien Garden	Goryokaku	Kenroku-en
3/17	0	0	0	0
3/18	1	0	1	0
3/19	0	0	0	0
3/20	0	0	0	0
3/21	1	0	0	0
3/22	0	0	0	0
3/23	0	0	0	0
3/24	3	0	0	0
3/25	0	0	0	0
3/26	0	0	0	0
3/27	0	4	0	0
3/28	0	4	0	0
3/29	3	2	0	0
3/30	5	2	0	0
3/31	1	3	0	0
4/1	4	1	0	0
4/2	2	1	0	0
4/3	0	1	0	0
4/4	2	0	0	2
4/5	1	0	0	2
4/6	0	0	0	0
4/7	0	0	0	0
4/8	0	0	0	0
4/9	0	0	0	1
4/10	0	0	0	0
4/11	0	0	0	1
4/12	1	0	0	2
4/13	0	0	0	0
4/14	0	0	0	1
4/15	0	0	1	0
4/16	0	0	0	0
4/17	2	0	1	0
4/18	4	0	0	0
4/19	1	0	0	1
4/20	0	0	1	0
4/21	1	0	1	0
4/22	0	0	0	0
4/23	0	0	0	0
4/24	0	0	1	0
4/25	0	0	2	1
4/26	0	0	3	0
4/27	0	0	1	0
4/28	0	0	0	0
4/29	0	0	0	0
4/30	0	0	0	0
Precision	0.0%	0.0%	100.0%	100.0%
Recall	0.0%	0.0%	14.3%	33.3%

using the proposed method. Confirmation of the flowering date and the full bloom date of each tourist spot is difficult using JMA data. Therefore, verification of this experiment was used to assess flowering and the best-time viewing the sights according to services or blogs, in addition to SNS of weather information companies [18] and public interest institutes [19]. The arrow representing the time to bloom from flowering was confirmed manually at each tourist spot.

Table 5 shows that the data of each tourist spot is very few, but one can confirm the differences of full bloom times for tourist spots. Even in the vicinity of each other like "Shinjuku Gyoen" "Rokugien", time difference can be seen. This result is different from estimation by JMA which depends on observation of specimen tree. The proposed method shows the possibility to estimate the best-time viewing date of each tourist spot. However, in the proposed method, the accuracy of extracting best-time viewing as a period is low. For that reason, it is a big task in the future as to estimation in regions and tourist spots with few tweets.

Table 6 presents experimentally obtained results for 2016. The notation is the same as that used in Table 5.

Table 6: Best time to see the estimated and tweet the number of tourist spot name to co-occurrence (2016)



Date	Shinjuku Gyoen	Rikugien Garden	Goryokaku	Kenroku-en
3/17	1	1	0	0
3/18	2	0	0	0
3/19	2	0	0	0
3/20	5	2	0	0
3/21	9	4	0	0
3/22	0	1	0	0
3/23	3	3	0	0
3/24	3	3	0	0
3/25	5	4	0	0
3/26	12	9	0	0
3/27	7	26	0	0
3/28	1	7	1	1
3/29	5	18	4	0
3/30	9	18	0	1
3/31	6	14	0	1
4/1	6	13	1	2
4/2	22	13	1	7
4/3	29	21	0	7
4/4	4	5	0	6
4/5	6	2	0	3
4/6	9	3	0	6
4/7	5	0	0	3
4/8	5	13	0	2
4/9	12	2	0	3
4/10	13	1	0	7
4/11	2	0	0	1
4/12	3	1	0	0
4/13	1	0	0	1
4/14	0	0	0	0
4/24	0	0	0	0
4/25	1	0	2	0
4/26	0	0	3	0
4/27	0	0	3	0
4/28	1	0	3	0
4/29	1	1	5	0
4/30	0	0	6	0
5/1	0	1	4	0
5/2	2	0	9	0
5/3	1	0	8	0
5/4	0	0	5	0
5/5	0	0	2	0
5/6	0	0	6	0
5/7	1	0	3	0
5/8	0	0	0	0
5/9	0	0	2	0
5/10	1	0	0	0
Precision	50.0%	100.0%	66.7%	66.7%
Recall	22.2%	27.8%	23.5%	12.5%

Similar to Tables 4 and 5 presented in earlier sections, 2016 had a longer best-time duration than that in 2015 because of lower temperatures after flowering. Table 5 shows that the data are increasing for each tourist spot. Therefore, the estimated best-time results obtained for 2016 using the proposed method tend to match the best-time viewing in 2015 indicated by the arrow. However, the best time estimation of the tourist resort including the optimum viewing time estimated using the proposed method increased the amount of data compared to 2015, so it was possible to confirm the improvement in accuracy, but further improvement of the method is necessary.

5 CONCLUSION

This paper suggested using Twitter to generate a useful approach to estimate the best time to present sightseeing information related to phenology observation. In the proposed method, we used a geo-tagged tweet containing the organism name of the target word to infer the optimal time to see flowers in Japan. The result of the cherry blossom experiment shows that the seasonal change of the tweet and the actual seasonal change are related to the estimate. Therefore, the proposed method presents the possibility to estimate the best time in the real world by observing the tweet related to the organism name. By using this, we are considering application to a system that can judge whether the phenomenon will become a tourist target when visiting sightseeing by checking whether the phenology is the best state in the area. Also, the granularity of the proposed method differs depending on the target word, region, and sightseeing spot. In this paper, we conducted experiments on sightseeing spots that co-occur with prefectures and target words. The results confirmed the possibility of displaying tourist information in real time for each area and sightseeing spot by estimating the optimum time using geotagged tweets. On the other hand, further consideration is needed on the estimation of the best time in areas and sightseeing spots where the number of data is small. Future research should verify that the proposed method is applicable to other organisms. Depending on the target word, there are many false positive cases, so in the future we will also consider methods of analyzing tweets contents. By extending the proposed method, we would like to connect to a system that allows travelers to obtain event information and disaster information on travel destinations in real time.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Nos. 16K00157, 16K16158, and a Tokyo Metropolitan University Grant-in-Aid for Research on Priority Areas "Research on Social Big Data."

REFERENCES

[1] Ministry of Internal Affairs and Communications, "2014 Communications Usage Trend Survey Results," http://www.soumu.go.jp/johotsusintokei/statistics/data/150717_1.pdf (7, 2015) (in Japanese).

[2] Twitter, "Twitter," https://Twitter.com/ (4, 2014).

[3] Ministry of Economy, Trade and Industry, "study of landing type IT tourism and attract customers service," http://www.meti.go.jp/report/downloadfiles/g70629a01j.pdf (3, 2007) (in Japanese).

[4] Manabu Okumura, "Microblog Mining," IEICE Technical Report NLC2011 −59 (2012) (in Japanese).

[5] J. Kleinberg, "Bursty and hierarchical structure in stream," In Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.1-25 (2002).

[6] K. Ochiai, and D. Torii, "Toponym Disambiguation Method for Microblogs Using Time-varying Location-related Words," IPSJ TOD, Vol.7, No.2, pp.51-60 (2014) (in Japanese).

[7] A. Kurata, K. Uehara, and J. Murai, "Auto situation detecting system using Twitter," The 75th National Convention of JPSJ, 1V-1, pp.97-98 (2013) (in Japanese).

[8] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: real-time event detection by social sensors," WWW 2010, pp.851-860 (2010).

[9] Florian Kunneman and Antal van den Bosch., "Leveraging unscheduled event prediction through mining scheduled event tweets," BNAIC2012, p.147 (2012).

[10] Ali Hurriyetoglu, Florian Kunneman, and Antal van den Bosch, "Estimating the time between twitter messages and future events," DIR-2013, pp.20–23 (2013).

[11] Hannah Tops, Antal van den Bosch, and Florian Kunneman, "Predicting time-to-event from Twitter messages," BNAIC 2013, pp.207-2014 (2013).

[12] Twitter Developers, "Twitter Developer official site, " https://dev.twitter.com/ (4, 2014).

[13] Y. Hashimoto, M. Oka, "Statistics of Geo-Tagged Tweets in Urban Areas(<Special Issue>Synthesis and Analysis of Massive Data Flow)," JSAI, Vol.27, No.4, pp.424-431 (2012) (in Japanese).

[14] National Agriculture and Food Research Organization, "simple reverse geocoding service," http://www.finds.jp/wsdocs/rgeocode/index.html.ja (4, 2014).

[15] MeCab, "Yet Another Part-of-Speech and Morphological Analyzer," http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html (9, 2012).

[16] Japan Meteorological Agency, "Disaster prevention information XML format providing information page," http://xml.kishou.go.jp/ (12, 2011).

[17] Japan Meteorological Agency, "observation of Sakura," http://www.data.jma.go.jp/sakura/data/sakura2012.pdf (3, 2015).

[18] Weathernews Inc., "Sakura information," http://weathernews.jp/koyo (9, 2015).

[19] Japan Travel and Tourism Association, "whole country cherry trees," http://sakura.nihon-kankou.or.jp/ (4, 2015).

(Received October 8, 2016)
(Revised February 16, 2017)

Masaki Endo

Masaki Endo earned a B.E. degree from Polytechnic University, Tokyo and graduated from the course of Electrical Engineering and Computer Science, Graduate School of Engineering Polytechnic University. He received an M.E. degree from NIAD-UE, Tokyo. He earned a Ph.D. Degree in Engineering from Tokyo Metropolitan University in 2016. He is currently an Assistant Professor of Polytechnic University, Tokyo. His research interests include web services and web mining. He is also a member of DBSJ, NPO STI, IPSJ, and IEICE.

Yoshiyuki Shoji

Yoshiyuki Shoji is a program-specific Researcher at Kyoto University, Japan. He earned a Ph.D. Degree in Informatics from Kyoto University in 2015. He worked as a researcher at Kyoto University in 2015, and worked as a program-specific Assistant Professor at Tokyo Metropolitan University in 2016. His research interests include social informatics, information retrieval, and web mining.

Masaharu Hirota

Masaharu Hirota earned his Doctor of Informatics degree in 2014 from Shizuoka University. Since April 2015, he has been working as an Assistant Professor at the National Institute of Technology, Oita College, Department of Information Engineering. His main research interests include geo social data, multimedia, and visualization. He is a member of ACM, IPSJ, and IEICE.

Shigeyoshi Ohno

Shigeyoshi Ohno earned M.Sci. and Dr. Sci. degrees from Kanazawa University and a Dr. Eng. degree from Tokyo Metropolitan University. He is currently a full Professor of Polytechnic University, Tokyo. His research interests include big data and web mining. He is a member of DBSJ, IPSJ, IEICE and JPS.

Hiroshi Ishikawa

Hiroshi Ishikawa earned B.S. and Ph.D. degrees in Information Science from The University of Tokyo. After working for Fujitsu Laboratories and becoming a full Professor at Shizuoka University, he became a full Professor at Tokyo Metropolitan University in April, 2013. His research interests include databases, data mining, and social big data. He has published actively in international refereed journals and conferences such as ACM TODS, IEEE TKDE, VLDB, IEEE ICDE, and ACM SIGSPATIAL. He has authored several books: *Social Big Data Mining* (CRC Press). He is a fellow of IPSJ and IEICE and is a member of ACM and IEEE.

Evaluation of Highly Available Safety Confirmation System Using an Access Prediction Model

Masaki Nagata[†1][†2][*], Yusuke Abe[†2], Misato Fukui[†2], Chihiro Isobe[†2], and Hiroshi Mineno[†1][**]

[†1]Graduate School of Science and Technology, Shizuoka University, Japan
[†2]AvanceSystem Corporation, Japan
[*] nagata@avancesys.co.jp
[**] mineno@inf.shizuoka.ac.jp

Abstract - A safety confirmation system provides a mechanism for sharing users' safety information in disasters, and is therefore required to operate reliably in the event of a disaster. Further, it is essential that the architecture is able to expand to accommodate additional resources during disasters because it is accessed by many users at such times. Increasing the appropriate resources during disasters necessitates the use of access prediction based on the access distribution during past disasters. Many conventional safety confirmation systems utilize a Relational Database (RDB) because the RDB structure is suitable for data management. However, because RDB has weak partition-tolerance characteristics it has availability issues. In this paper, we propose a method that improves the partition-tolerance using multiple servers, and an access prediction method that utilizes lognormal distribution to predict access to safety confirmation systems during disasters. The proposed method also employs a distributed database system with multiple servers and access prediction is carried out using a plurality lognormal distribution that depends on the time at which a disaster occurs. The results of evaluations conducted indicate that the proposed method improves availability and allocates the appropriate resources for access distribution during disasters.

Keywords: access prediction, lognormal distribution, distributed database system, load balancing, safety confirmation system

1 INTRODUCTION

The ability to share safety information with users during disasters that result in serious damage and life-threatening danger, such as the Great East Japan Earthquake of 2011 and the 2016 Kumamoto Earthquake, is important because the early collection and disclosure of user safety information can save many lives. A safety confirmation system provides a means of sharing information with users during disasters [1]. A safety confirmation system is a web system that collects and presents safety information during disasters from and to users registered in the system. For example, the disaster bulletin board of a telecommunications carrier, Google Person Finder [2], and J-anpi [3] can crossover and collectively search the safety information they each have available. These systems are suitable for implementation using a web system, because a web system is accessible by PC and smartphone for reporting and presenting safety information. In addition, the system operation infrastructure can be outsourced to a cloud vendor that has disaster countermeasures rather than a single company's on-premises assets because a safety confirmation system is required to operate continuously during a disaster. However, migrating a system to the cloud environment is problematic.

The first issue is that of distributed data management for system redundancy. Because a safety confirmation system is required to operate continuously and reliably during disasters, its data management has to include strong partition-tolerance that enables alternate operation on another server when the primary server is down. Fu [4] proposed a method that improves availability using a redundant server to configure the system. In addition, we previously proposed a general safety confirmation system with global redundancy; that is, with servers in multiple regions, overseas as well as domestic. The use of multiple servers enables inevitable operation as a distributed system. Moreover, the study of conventional safety system [5] [6] that contains the author's previous studies is using an RDB for data management. However, that conventional safety confirmation system uses an RDB for data management, which poses a problem as an RDB has weak partition tolerance.

The second issue is that of adjusting the number of servers in accordance with the access situation. Because a safety confirmation system has very high access traffic during disasters and very low traffic when there is no disaster, the number of servers utilized should vary accordingly in order to reduce the operating cost. Access to the safety confirmation system increases during disasters; hence, the ability to determine the number of servers suitable to accommodate access traffic during a disaster is important. We obtained an understanding of the tendency of access traffic during disasters by analyzing access distribution during past disasters. As a result, real access traffic was found to exhibit a lognormal distribution. Consequently, we previously proposed an access prediction model that uses lognormal distribution [7]. The access prediction model showed that the cost of using additional servers can be reduced by allocating an appropriate number of servers for access distribution that varies with time during a disaster. However, access prediction is problematic in that it depends on the disaster situation. Thus, to overcome these issues, in this paper we propose a method that uses a distributed database with multiple servers and access prediction using a plurality lognormal distribution. We demonstrate the effectiveness of prototype safety confirmation system with these functions implemented.

ISSN1883-4566

2 RELATED WORK AND ISSUES

2.1 Safety Confirmation System

Work related to safety confirmation systems has been reported in various fields, e.g., information collection and sharing, network communication, and web systems. As regards information collection and sharing, Ishida et al. [8] proposed a safety information system that gathers and shares refugee information between different evacuation centers set up by each local government during a disaster. Registration of refugee information is accomplished using a personal IC card issued to each user and a reader. This is in consideration of children and older people inexperienced with ICT equipment. As regards network communication, Wang et al. [9] proposed a system that uses the AODV protocol to enable communication between users using smartphones. The system enables reliable transmission of safety information using node-to-node communication when the communication infrastructure is damaged or usage of communication resources is restricted.

The subject of this study is a general safety confirmation web system. The process followed by a safety confirmation system is as follows (Fig.1). First, the meteorological information service provides information about the occurring disaster to the safety confirmation system. Then, during the disaster, the safety confirmation system sends an e-mail to prompt users for safety confirmation. Next, users who receive the e-mail report their safety information to the safety confirmation system. Finally, users share their safety information with each other. Yuze and Suzuki [10] proposed relocating safety confirmation systems running on on-premises equipment to the cloud environment to improve service availability and to ensure sustained operation should the on-premises environment be adversely affected during the disaster. Echigo et al. [11] proposed load balancing and redundancy by mirroring using multiple servers to improve robustness. Thus, web systems have generally been used for information management in communication and information gathering related work. Therefore, it is clear that sustainable operation of the web system infrastructure is important to achieve effective overall safety information management during disasters.

2.2 Issues: Distributed Data Management

In a conventional safety confirmation system, data are managed using an RDB because the Create, Read, Update, Delete (CRUD) operation of each attribute data with a key such as user ID is suitable for managing users' safety information and department data. However, access traffic to each piece of attribute data is usually low; much of the access is the safety report during the disaster. Safety report access is an Update operation to update users' safety information data. The RDB sharding technique for dispersed access using multiple servers is an advantage but RDB has problems such as data search complexity and change of the ID numbering of the hash function associated with the data scale. In addition, RDB is not suitable for distributed systems because it is vulnerable to partition-tolerance of the CAP theorem. By contrast, a safety confirmation system should use distributed data management that runs on another server when the main server is down because continuous operation is essential. Therefore, it is necessary to improve availability using multiple servers and distributed data management that is able to manage high-volume access traffic during disasters.

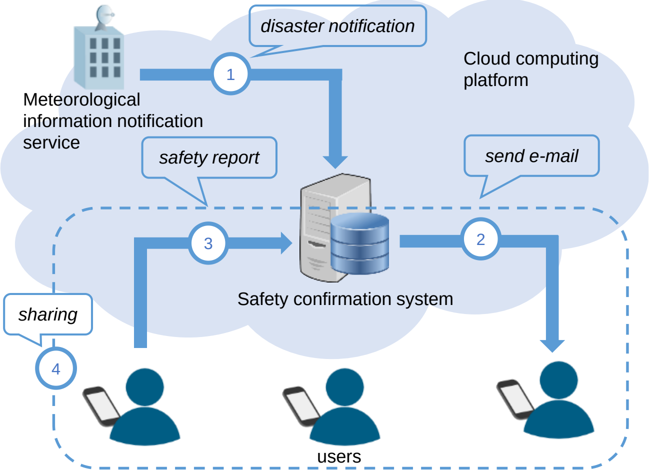


Figure 1: Flow of the safety confirmation system

2.3 Issues: Number of Servers in Accordance with the Situation

The cost of safety confirmation systems, which differs depending on the number of users accessing the system normally and during disasters, can be reduced by operating the number of servers in accordance with the access situation. The most simplistic resource management is to continue running the system on a large number of servers, regardless of the situation. However, the smaller amount of access traffic during when there is no disaster means that the continuous operation of many servers at all times results in surplus resources, and, consequently, surplus costs. Therefore, if the required number of servers can be ensured to be in accordance with the access situation, this would be ideal for the resource management of the safety confirmation system. It would reduce the cost when there is no disaster, when the amount of access traffic is small. Moreover, calculating and allocating the appropriate number of servers before access concentration is desirable to avoid impairing user convenience when the response performance decreases. Calculation of a suitable number of servers in accordance with the access situation necessitates prediction of the access distribution to the system during disasters. In our previous study, we proposed an access prediction model that uses a lognormal distribution to predict access to the safety confirmation system during disasters. However, this approach is problematic as the use of a single lognormal distribution to model access prediction is difficult. This is because the access distribution trend to the system was found to differ according to the time at which a disaster occurs. Therefore, it is necessary to calculate the number of servers by selecting the appropriate access prediction model in accordance with the disaster occurrence time.

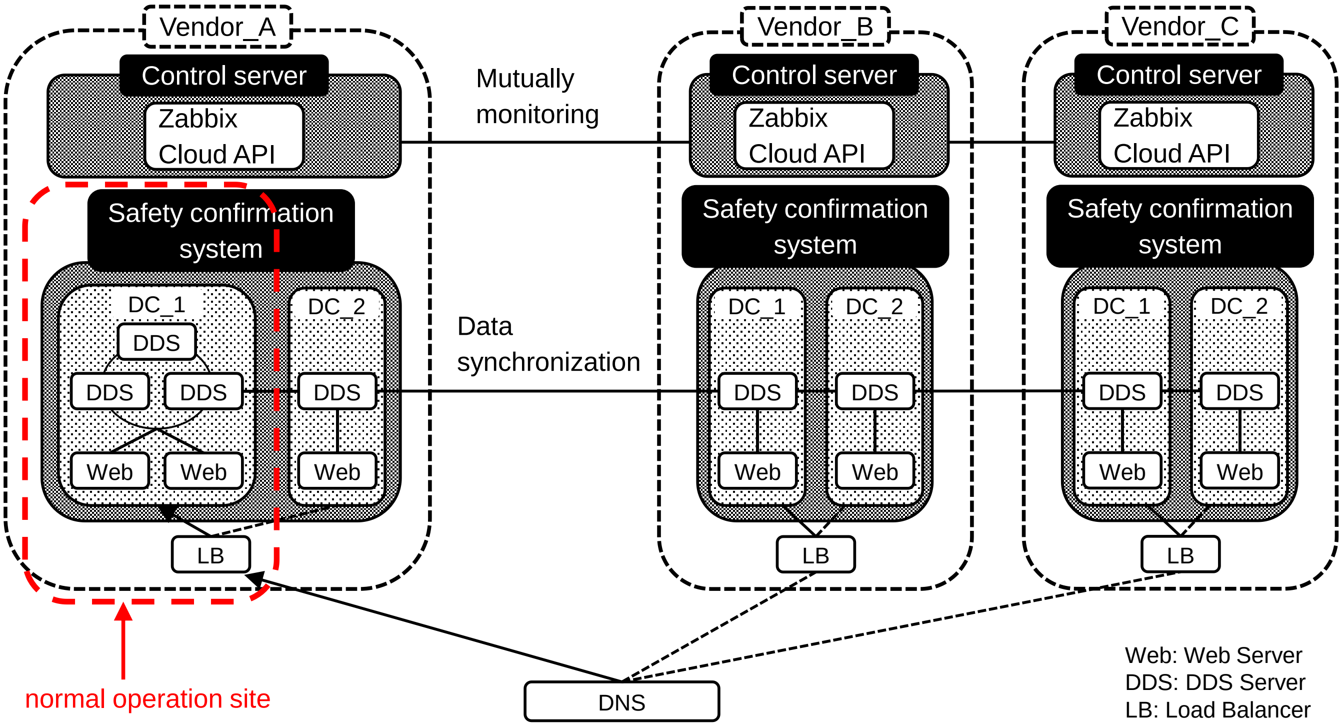


Figure 2: Architecture of the proposed system

3 PROPOSED SYSTEM

3.1 System Overview

The proposed system operates in an intercloud environment using three cloud vendors (Fig.2): vendor A, Amazon Web Services (AWS) [12]; vendor B, Microsoft Azure [13]; and vendor C, Cloudn [14]. DC_1 and DC_2 are the service provision regions of the vendors, each of which has multiple data centers. The system availability was improved by the monitoring from each vendor. During normal operation of the system, all accesses are directed to vendor A and vendor A is in charge of load balancing against increased access during a disaster. Vendor B and C are backup sites for vendor A. When a failure occurs, failover is accomplished by changing the access destination to vendor B (or C). The safety confirmation system and control server is deployed to each vendor. The safety confirmation system consists of a web server and a Distributed Data System (DDS). The DDS is a mechanism for distributed management of the data in cooperation with multiple servers. Distributed data management is implemented by arranging a plurality DDS node to each vendor. Data synchronization is carried out by using a data replication function. The control server uses the access prediction model to calculate the appropriate number of servers required during the disaster, and to scale out the web server for the safety confirmation system in order to conduct load balancing. In addition, it monitors each vendor and uses Zabbix [15] to perform failover when failures occur. Incidentally, the data synchronization of each vendor and the recovery flow in the event of failure are not discussed in this paper as those have already been expounded on in [7].

An overview of the safety confirmation system is shown in Fig.3. The figure depicts operation by multiple customers to share the server resources and usage by registered users. A summary of the operation after the occurrence of the disaster that corresponds to the region and the earthquake intensity threshold set by the customer unit is sent by e-mail to promote the safety report to the target users. Figure 3, for example, shows that an earthquake of intensity five upper occurred in Tokyo and Kanagawa, and that the number of target users is 15,700 among customers A, B, and D. Specifically, the number of target users of the proposed system changes according to the scale of the disaster. The system performs access prediction and load balancing in accordance with the number of target users.

Figure 4 shows the load balancing flow using the access prediction model. Inserting an additional server is called a scale-out operation, whereas reduction is a scale-in operation. The scale-out operation is not executed if the server is acceptable with the normal configuration of servers for the number of target users at the disaster; if unacceptable, scale-out executes with the appropriate number of servers based on the access prediction model. Load balancing is executed by scaling-out the web server in units of two servers, one for each of the two locations in "Vendor A: DC_1: AWS." Thus, it will add two, four, six–an even number of web servers. The proposed system equally distributes the load by utilizing the same number of web servers in each data center via the load balancer. Moreover, an e-mail is sent to target users following scale-out completion to avoid access concentration before the construction of a load balancing environment.

In order to execute a scale-out, it is necessary to ascertain the load point of the system. This is because it is possible to improve the processing power by adding a server when it accepts a certain load in terms of system resources.

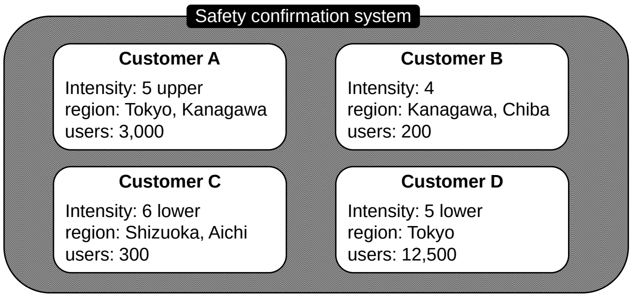


Figure 3: Multiple customer operation

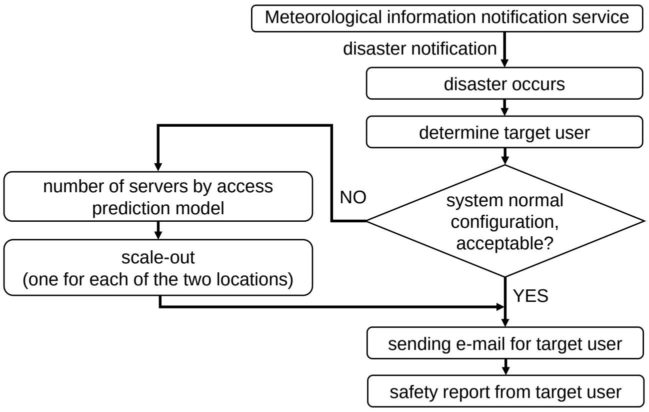

Meteorological information notification service
disaster notification
disaster occurs
determine target user
number of servers by access prediction model
NO
system normal configuration, acceptable?
scale-out (one for each of the two locations)
YES
sending e-mail for target user
safety report from target user

Figure 4: Flow of load balancing

Therefore, it is necessary to ascertain the load point of the system resources of the safety confirmation system during a disaster. Access to the system during a disaster accounts for more than 90% of the safety report accesses, based on access logs. Thus, the load point of the system is the safety report access concentration at the disaster. We measured the resource consumption of the load point using JMeter [16] to create a test scenario for safety report access. JMeter is a set of evaluation tools that enables a target system to be accessed via the web. Figure 5 shows the results of measuring each of the resources by changing access to the safety report every 10 minutes. The web server used AWS EC2 [17] t2.small and the DB server used EC2 c3.xlarge. Figure 5 shows that the web server CPU usage increased significantly with increasing safety report access traffic and each of the resource loads is considerably less than the web server CPU usage. This result indicates that the load point of the safety confirmation system increases web server CPU usage because of the safety report access concentration at the disaster. Therefore, the proposed system performs scale-out and scale-in by monitoring web server CPU usage. It should be noted that, strictly speaking, the database server was assigned a load, but this paper only targets the web server to simplify the explanation.

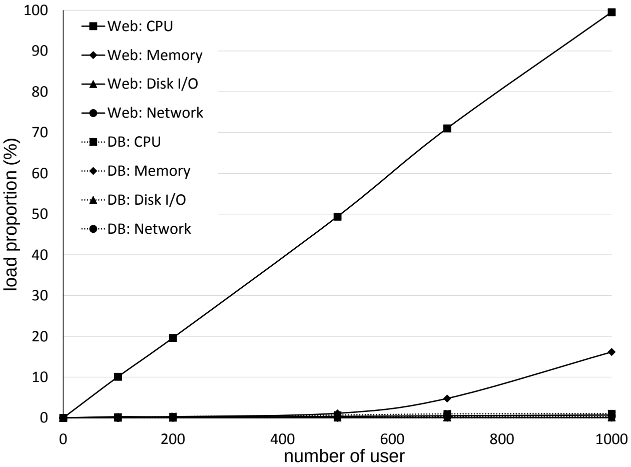


Figure 5: Web, DB resource usage proportion

Table 1: EC2 instance types: UnixBench results

Instance Type	vCPU	Memory (GiB)	System Benchmarks Index Score	Costs ($,hour)
t2.small	1	2	1702.30	0.034
t2.medium	2	4	2536.00	0.068
t2.large	2	8	2537.20	0.136
m3.medium	1	3.75	848.90	0.077
m3.large	2	7.5	1858.60	0.154
m3.xlarge	4	15	2945.00	0.308
m4.large	2	8	2025.00	0.14
m4.xlarge	4	16	3132.80	0.279

Server types and number of servers to be used in scale-out and scale-in are decided based on single server processing power. Murta and Dutra [18] modeled the resource management of an entire system from the benchmark result of a single server. In this paper, we calculate the appropriate number of servers based on the processing of a single server to determine the access traffic obtained in the prediction. The access processing power of this study is determined using the safety report access that can be processed per unit time. Each vendor has a variety of server types; measurements were conducted with respect to AWS, vendor A, which is the main one that performs load balancing. This was measured to clarify the relationship between the access processing power and each EC2 instance type. The measurement method is the same for each vendor. Table 1 shows the UnixBench [19] measurement results for a general-purpose EC2 instance type. The overall performance index of UnixBench is provided by the “System Benchmarks Index Score.” Table 1 shows that the “System Benchmarks Index Score” increased with EC2 “vCPU.” However, the “System Benchmarks Index Score” is not simply doubled when “vCPU” is doubled. Thus, the cost performance is higher for one vCPU than for two vCPUs. Therefore, the proposed system adopted t2.small from among the available “vCPUs,” considering the cost per hour and result of the “System Benchmarks Index Score.” Incidentally, AWS EC2 defines the baseline of CPU usage for the t2 series, including t2.small. If the CPU usage is above the baseline, the state becomes burst. Burst is a state in which CPU performance is temporarily reduced; it is able to continue by consuming the AWS CPU credits. If the CPU credits are exhausted, CPU performance cannot exceed the baseline. In this study, the processing power of one server was determined by the number of safety report accesses at a CPU usage of 20%, the baseline for t2.small, not considering the processing power of the burst. CPU usage at 20% of t2.small is able to process 200 safety report accesses in 10 minutes, as shown in Fig.5. Moreover, the processing power

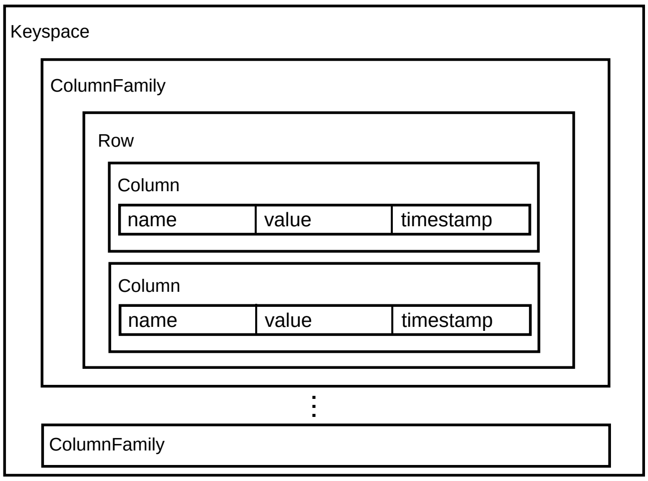


Figure 6: Data structure of cassandra

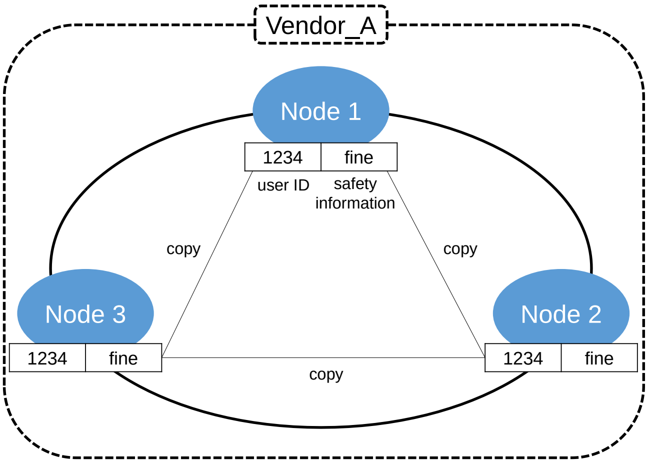


Figure 7: Node and replication factor

of the system with normal configuration is 400 safety report accesses in 10 minutes, because there are two t2.small instances for each web server.

3.2 Distributed Data Management Using Cassandra

Data management of the proposed system that converts the RDB data schema is desirable, because conventional safety confirmation systems use RDB for data management. Therefore, for data management of the proposed system Cassandra [20], which requires a schema definition among the distributed data management systems, was chosen. Cassandra is a NOSQL distributed data management system that also has excellent writing characteristics [21]. The data management structure of Cassandra is the Key-Value (KVS) method for managing a unique label (Key) for the data (Value). Cassandra is also a column-oriented NOSQL system. The column-orientation is obtained in an advanced simple KVS manner. It allows for multiple management of a set of Key and Value, which is referred to as a Column in Row, whereas simple KVS is managed in a one-to-one relationship between the Key and Value. The data structure of Cassandra is shown in Fig.6. The data units of Cassandra and RDB correspond in the following manner. Keyspace is database,

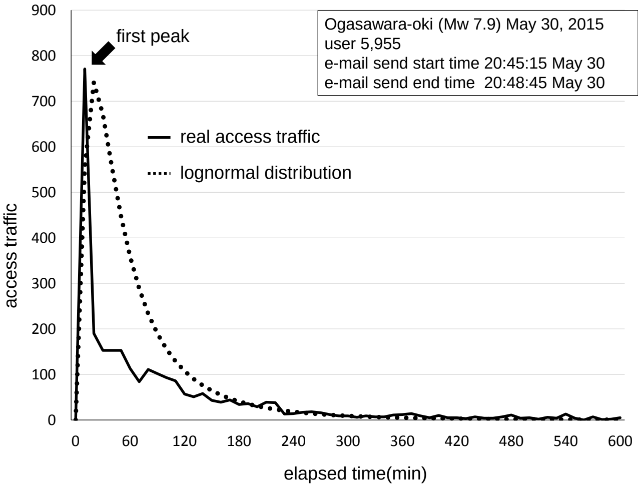


Figure 8: One-peak access traffic and lognormal distribution

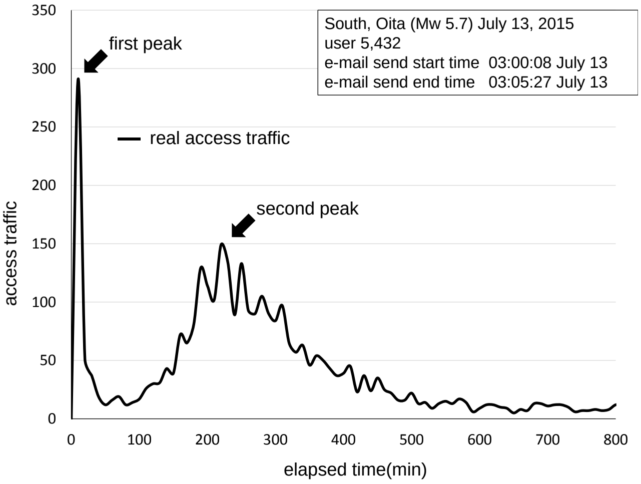


Figure 9: Two-peak access traffic

ColumnFamily is table, and Row is record. Porting of the data management is more easily done than other NOSQL systems because this structure is similar to the user management schema in RDB in the safety confirmation system.

Cassandra is typically operated on a cluster using multiple servers, rather than a single server. Operation in a cluster configuration enables continuous operation and improves availability as another node is alternatively operated when a node goes down. As shown in Fig.7, the proposed system operates in three nodes on vendor A, with Replication Factor (RF) = 3. Nodes from vendors B and C are used as backup. The RF is the total number of copies of the data. As shown Fig.7, in the case where RF = 3, the data have the user ID in Key and the safety information in Value to keep a copy of the data in three nodes. The proposed system improves the availability by copying the data to all nodes because the number of nodes is three RF = 3. It prepares for disaster recovery using the multi-data center capabilities of Cassandra so that nodes of vendor A and nodes of vendor B and C are involved in the automatic replication. Cassandra improves the availability by setting the number of nodes and RF properly. Therefore, it is suitable for data management of systems that require continuous operation, such as the safety confirmation system.

3.3 Access Prediction Model

3.3.1 Characteristic Access Distribution of the Safety Confirmation System

The prediction of access requires an understanding of the characteristics of access distribution with respect to the safety confirmation system. Figure 8 shows the access distribution during a disaster. The access traffic during the disaster reaches a peak a short while after the initial e-mail is sent by the safety confirmation system, and then decreases with time. Moreover, Fig.8 shows the lognormal distribution and access traffic during the disaster. To model the counting data, Poisson distribution is usually utilized. However, we propose using a lognormal distribution to predict access to the safety confirmation system during disasters, because we previously confirmed a certain normality via a normality test of access distribution in a disaster in our previous study [7]. When using this method, access is in accordance with a lognormal distribution with decay period from peak, and we achieved the expected effect in calculating the appropriate number of servers for access prediction; however, depending on the disaster situation, a single lognormal distribution is problematic.

Figure 9 shows the access traffic of a disaster that occurred at 3:00 at night. Figure 9 shows two peaks, with the first peak being immediately after the occurrence, and the second peak a few hours after the occurrence. The second peak is reached in the morning and reflects human activity time. This denotes that users who were sleeping during the disaster only accessed the system after awakening. Therefore, our proposed access prediction method uses a plurality lognormal distribution for access distribution consisting of two peaks.

3.3.2 Suitability with the Mixed Lognormal Distribution for Two Peaks

Access prediction is carried out by modeling the access trend distribution analysis using data from past disasters. Our previous study entailed access prediction with a single lognormal distribution model for a one-peak disaster, as shown in Fig.8. Lognormal distribution is defined as in Eq. (1), mode M is Eq. (2), expected value E is Eq. (3), where μ is the expected value of the normal distribution, and σ is the standard deviation of the normal distribution. Each parameter of the lognormal distribution is determined by analyzing past disasters and disaster drill data. As detailed information can be found in [7], only an outline is given here. Mode M is the time with the largest number of accesses and has fixed value of *20*. Expected value E is the time of the average number of accesses and is calculated from the relation between the number of target users *TU* and mode M. Further, solving the simultaneous equations of Eq. (2) and Eq. (3) results in Eq. (4) and Eq. (5). Then, substituting M and E into Eqs. (4) and (5) gives μ and σ, the parameters in Eq. (1). Equation (6) is the access prediction model, which is used to predict the access number *AN* at a time of x minutes. A is a coefficient of Eq. (1) for matching the peak access according to the number of target users *TU*.

$$f(x) = \frac{1}{x\sqrt{2\pi}\sigma} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2} \right) \tag{1}$$

$$M = \exp(\mu - \sigma^2) \tag{2}$$

$$E = \exp(\mu + \frac{\sigma^2}{2}) \tag{3}$$

$$\mu = \frac{(\ln(M) + 2 * \ln(E))}{3} \tag{4}$$

$$\sigma^2 = \frac{2 * (\ln(E) - \ln(M))}{3} \tag{5}$$

$$AN = A * f(x) \tag{6}$$

Toriumi et al. [22] conducted an analysis using the mixed lognormal distribution model for the concentration of multiple tweets from Twitter [23] during a disaster. This study examined the suitability of applying the mixed lognormal distribution for access distribution with two peaks with reference to previous research. The mixed lognormal distribution represented in Eq. (7) is based on Eq. (1). Moreover, it has two lognormal distributions considering its adaptation of two peaks. *f(x)* is the first peak lognormal distribution, *g(x)* is the second. α and β are weighting coefficients for the cumulative probability density of *F(x)*. It shall has the relation $\alpha+\beta=1$.

$$F(x) = \alpha f(x) + \beta g(x) \tag{7}$$

The determination of α and β was accomplished by calculating from the ratio of each distribution. T is the total reported number of each distribution; S is the reported number of the first distribution; and R is the reported number of the second distribution. α and β are presented in Eqs. (8) and (9), respectively.

$$\alpha = S / T \tag{8}$$

$$\beta = R / T \tag{9}$$

Note that we used two values of α and β because two distributions were targeted this time. Hence, multiple distributions can be handled by considering them like that in Eq. (10). N is the number of coexisting distributions, and c denotes the weighting coefficients for the cumulative probability density. In this case, c_1 is α, c_2 is β, $f_1(x)$ is $f(x)$ and $f_2(x)$ is $g(x)$. In addition, c has the condition of Eq. (11).

$$F(x) = \sum_{i}^{N} c_i * f_i(x) \tag{10}$$

$$\sum_{i}^{N} c_i = 1 \tag{11}$$

The basic formula of Eq. (7) only strictly represents the probability distribution. Thus, the weighting coefficient must be further determined to represent the access distribution. The access distribution is represented by Eq. (12):

$$H(x) = A * \alpha f(x) + B * \beta g(x) \tag{12}$$

where *A* is the first peak adjustment coefficient and *B* is the second peak adjustment coefficient. Similar to that in Eq. (10), handling multiple distributions is possible by considering them like that in Eq. (13). *D* is the weighting coefficient to represent the access distribution. In this case, D_1 is *A* and D_2 is *B*.

$$H(x) = \sum_{i}^{N} D_i * c_i * f_i(x) \tag{13}$$

Substituting *x_1* and *x_2* of the first and second peaks, respectively, into Eq. (12), and solving the simultaneous equations enables *A* and *B* to be determined.

4 IMPLEMENTATION AND EVALUATION

4.1 Implementation

Table 2 shows the cloud vendor and the instance type to be used in the proposed system. Table 3 shows the cloud API of each vendor and each of the control servers of each vendor. Access to cloud resources is carried out using this API, and it also controls other vendors not only its own vendor. Table 4 shows each server in the system environment. Zabbix is run on each vendor's control server to perform fault detection. For example, if a fault is detected on DC_1 of vendor A, vendor B or C can change the access destination to DC_2 using the aws-cli by changing the setting of the load balancer of vendor A. The web server of the safety confirmation system has machine images of the source code and the OS with the same content. Thus, during failure or scale-out, the machine image is started under the load balancer using the API in Table 3. The machine image is an image of the activation information of the server that includes the middleware (the database management system, etc.), the binary code of the application software, device drivers and OS, and so on. Servers with the same configuration can be rapidly duplicated using a machine image. The Cassandra node also maintains a machine image in the same manner as the web server. At this point, safety information data are not included in the machine image. Cassandra is not necessary at the same time as the machine image safety information data; it only performs data synchronization connected to the cluster with the participation at the time of start-up. Cassandra becomes operational when ready after data synchronization; the status becomes "Up Normal (UN)," indicating normal operation.

4.2 Evaluation of Distributed Data Management

In the evaluation of the distributed data management in Cassandra, it was confirmed that the safety information data can be retrieved correctly in an environment that has a stopped DC_1 single node from vendor A. In addition, vendor A was intentionally stopped, and alternative operation by vendors B and C at the backup sites was confirmed.

Table 2: Cloud vendors and instance types

Vendor_A	AWS	t2.small
Vendor_B	Azure	Standard A1
Vendor_C	Cloudn	Plan v1

Table 3: Cloud API

AWS	aws-cli 1.10.56
Azure	Azure cli 0.10.3
Cloudn	Cloudn SDK for Ruby 0.0.1

Table 4: System environment

Safety Confirmation System Web Server	CentOS 6.5 Apache 2.2.15
Safety Confirmation System DB Server	CentOS 6.5 Cassandra 2.0.6
Control Server	CentOS 6.5 Zabbix 2.4.7

This is because other nodes also hold the data. As shown in Fig.10, Cassandra copies data to other nodes when data are written to a node from the client. Thus, even when one node stops, the system can operate on other nodes. Therefore, we confirmed that the system can operate with other nodes when a node is stopped. The availability improvement of the proposed system was also confirmed in this manner. Moreover, after the node was stopped, it was confirmed that safety information data are correctly acquired by adding a new node. Node addition and preparation took about 90 seconds. Figure 11 gives a performance comparison of RDB and Cassandra. The evaluation environment is RDB and Cassandra is one respectively, and 12 unit web servers were placed under the DC_1 load balancer of vendor A. We conducted numerous safety report accesses using JMeter to evaluate the environment. The RDB used was PostgreSQL8.4.12. We measured the CPU utilization and throughput of the web server and the DB server in the case where the safety report access increased from zero to 5,000 per 10 minutes. Measurement of CPU utilization was achieved using the "sar" command. Measurement of the throughput was achieved using JMeter. Although CPU utilization of Cassandra is slightly lower in the throughput value of the same degree,

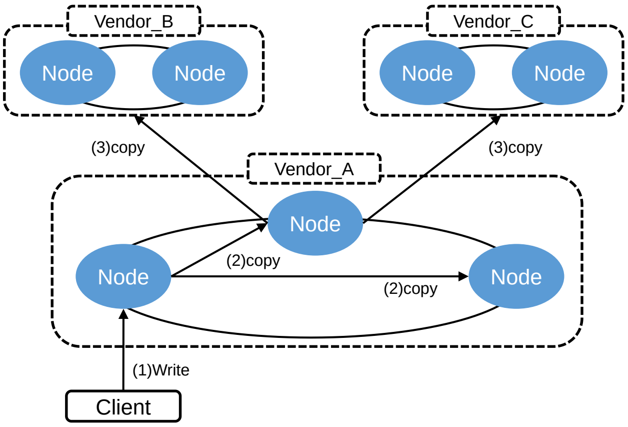


Figure 10: Flow of the data copy operation

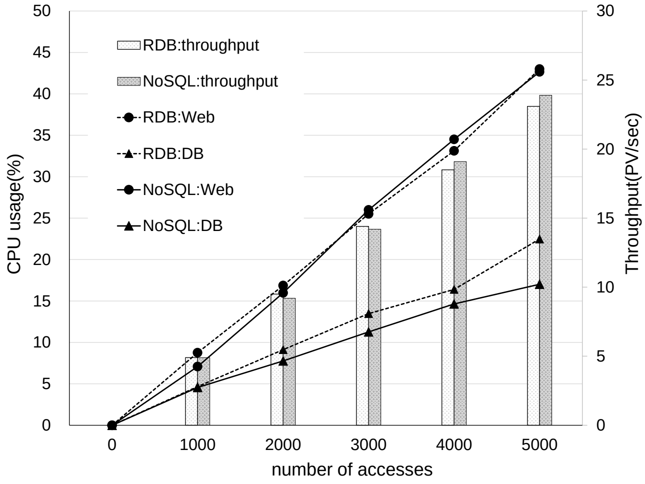


Figure 11: Performance comparison of RDB and NOSQL

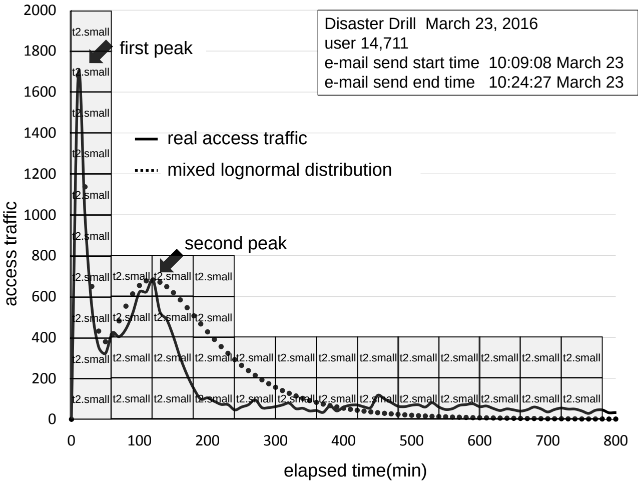


Figure 13: Disaster drill

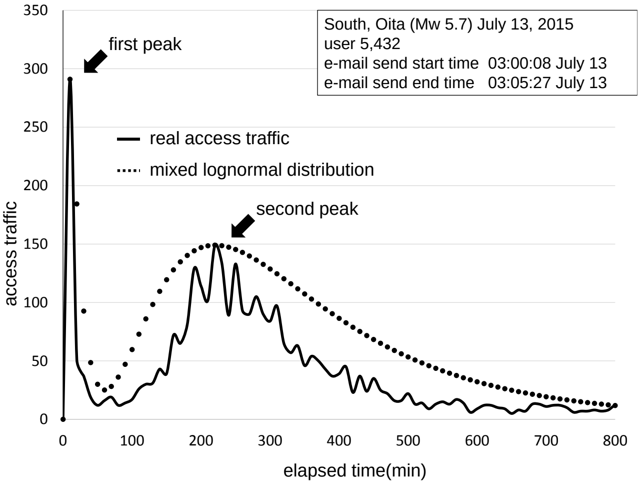


Figure 12: Oita earthquake

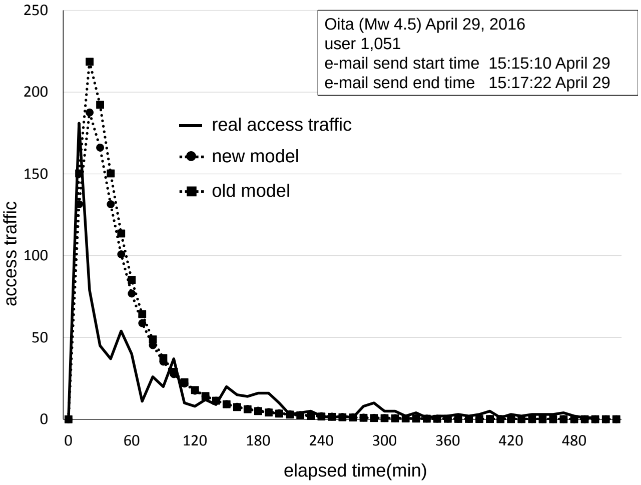


Figure 14: Accuracy of the model during Oita disaster

as the number of accesses increase, there is virtually no difference between both Cassandra and RDB. However, Cassandra is superior in terms of availability using multiple nodes compared with the RDB because Cassandra usually does not operate in a single node.

4.3 Evaluation of Suitability for Access Distribution

The suitability of the proposed method was evaluated for access distribution. The evaluation was carried out using 10 minutes of access distribution during the Oita earthquake, shown in Fig.9 (Fig.12), and the disaster drill in a company (Fig.13). A safety confirmation system has a similar load point in a disaster and a disaster drill because access is concentrated from the start of the disaster drill. Therefore, disaster drill data were also used to evaluate the proposed method. Moreover, the disaster drill is similar to the access distribution at midnight. During the disaster drill, the first peak occurred after the start of the drill when the notification e-mail was sent in the morning, and there was a second peak during the lunch break. The number of target users of the Oita earthquake (Fig.12) was 5,432 people. At the first peak, 291 users accessed the system 10 minutes after the occurrence, and the second peak resulted from 149 users accessing the system 220 minutes after the occurrence. Substituting x_1=10, x_2=220 into Eq. (12), and solving the simultaneous equations produces the following results A=50,443.9, B=64,225.3. Then, σ and μ are determined from M and E, for the first and second distribution, respectively, which become the distribution curve of the proposed method, as shown in Fig.12. The number of target users of the disaster drill (Fig.13) is 14,711 people, and the first peak occurs when 1,680 users access the system 10 minutes after the occurrence and the second peak when 681 users access the system 120 minutes after the occurrence. Substituting x_1=10 and x_2=120 into Eq. (12) and solving the simultaneous equations produces the result A=154,997.0, B=175,775.9. The result shows that the peak of the proposed method is consistent with the access distribution because the known peak value of the past disaster is fitted to Eq. (12). However, the proposed method is also generally acceptable for subsequent distribution. This paper evaluated the suitability of Fig.9 based on the mixed lognormal distribution case having two peaks. However, it is necessary to evaluate with many cases in the future.

It shows the calculation of the number of servers for the calculated access distribution using the proposed method. The number of servers was calculated based on the processing power required to access one server with an access distribution of 1-hour increments, because EC2 is charged on an hourly basis. Figure 13 shows that there is access of up to 1,680/10 minutes during 0–60 minutes. The processing power of t2.small is 200 safety report accesses in10 minutes; therefore, 0–60 minutes is for 10 servers. We also calculated the number of servers in the same manner.

4.4 Accuracy of the Model by the Number of Sample Data

Each of the parameters used in the proposed model are determined by statistical analysis using data from past disaster and disaster drill data. As an example, *E* is determined by the approximation equation using the relation of the *TU* and *D*. *D* is difference between *E* and *M*. The coefficients to be granted to the lognormal distribution are calculated from the ratio of *TU* and the number of peaks. Therefore, a large amount of data for use in the analysis is expected to improve the accuracy of the proposed method. The access prediction model of a single lognormal distribution was a comparative evaluation of the amount of data, which is less in the old model and more in the new model. The old model was used until the summer of 2015, the amount of data was 14, and the amount of data in the new model to which data was subsequently added, was 29. Figure 14 shows a trace of the disaster data, whereas Fig.15 shows the trace of the disaster drill data. Compared to the old model, both new models are well suited for real access distribution, and it is seen that the accuracy of the new model is improved. The proposed method uses mixed lognormal distribution, which only fits the access distribution to disaster data and disaster drill data. However, as shown with the single lognormal distribution model in Fig.14 and 15, hereafter, data can grasp the tendency of each of the parameters to be given to the model by collecting, and can be expected to build an access prediction model.

5 CONCLUSION

In this paper, we proposed a distributed database system that uses multiple servers to improve the availability of safety confirmation systems and an access prediction method that uses a lognormal distribution to predict the concentration of access to the safety confirmation system during a disaster. The DDS using a plurality of Cassandra nodes achieved high availability to continue the operation even when a single node has stopped. The proposed method uses a mixed lognormal distribution and indicates that it is possible to compute access prediction for an access distribution with two peaks resulting from the occurrence of a disaster situation.

Future challenges include the construction of the access prediction model using a mixed lognormal distribution and improving the accuracy. The mixed lognormal distribution model showed the suitability of the extent to which access distribution was allowed during the past disaster and disaster drill. However, this is a poor basis for relevance because the amount of sample data for access prediction was small. A parameter of the model is determined based on past empirical rule with simple consideration. However, using the least squares method or the maximizing likelihood method, it can be expected to further improve the model accuracy. In addition, there is a need for verification and evaluation of the Poisson distribution as well as the normal distribution. In the future, we plan to improve the accuracy and modifications of the proposed method by collecting an additional amount of disaster data.

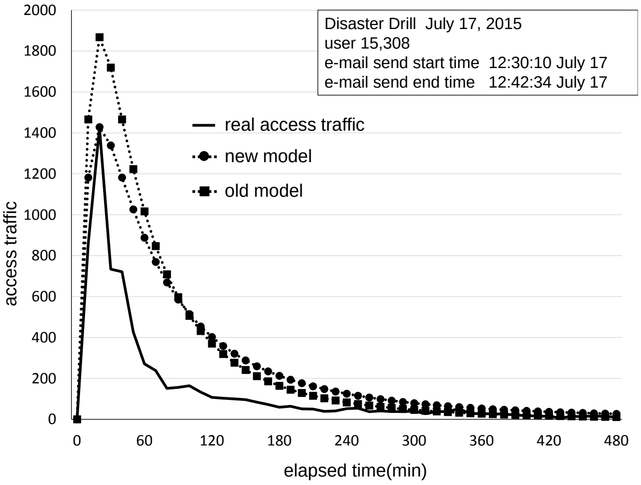


Figure 15: Accuracy of the model during disaster drill

ACKNOWLEDGMENTS

We are grateful to Associate Professor Takahiro Hasegawa at the Center for Information Infrastructure, Shizuoka University who developed the safety confirmation system that became the basis for the proposed system. In addition, we are grateful to Kunihiko Murayama, president of AvanceSystem Corporation.

REFERENCES

[1] T. Hasegawa, H. Inoue, and N. Yamaki, “Development of a low running cost and user friendly safety information system,” Journal for Academic Computing and Networking, No. 13, pp. 91–98 (2009). (in Japanese).

[2] Google Person Finder, https://google.org/personfinder/global/home.html?lang=en, (2016).

[3] J-anpi, http://anpi.jp/top, (2016) (in Japanese).

[4] S. Fu, “Failure-aware construction and reconfiguration of distributed virtual machines for high availability computing,” IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 372–379 (2009).

[5] S. Kajita, Y. Ohta, S. Wakamatsu, and K. Mase, “Stepwise Development of a Survivor Confirmation System for a Higher Educational Institution and Its Production Use,” IPSJ Journal, Vol.49, No.3, pp.1131–1143 (2008). (in Japanese).

[6] H. Echigo and Y. Shibata, “Performance Evaluation of Large Scale Disaster Information System over Japan Gigabit Network,” IEEE 22nd International Conference on Advanced Information Networking and Applications, pp. 1101-1106 (2008).

[7] M. Nagata, Y. Abe, I. Kinpara, M. Fukui, and H. Mineno, “A proposal and evaluation of a global redundant safety information system based on access prediction model,” IPSJ Transactions on Consumer Devices & Systems, Vol. 6, No. 1, pp. 94–105 (2016). (in Japanese).

[8] T. Ishida, A. Sakuraba, K. Sugita, N. Uchida, and Y. Shibata, “Construction of safety confirmation system in the disaster countermeasures headquarters,” Eighth International Conference on 3PGCIC. IEEE, pp. 574–577 (2013).

[9] J. Wang, Z. Cheng, I. Nishiyama, and Y. Zhou, “Design of a safety confirmation system integrating wireless sensor network and smart phones for disaster,” IEEE 6th International Symposium on Embedded Multicore Socs, pp. 139–143 (2012).

[10] H. Yuze, and N. Suzuki, “Development of cloud based safety confirmation system for great disaster,” IEEE 26th International Conference on Advanced Information Networking and Applications Workshops, pp. 1069–1074 (2012).

[11] H. Echigo, H. Yuze, T. Hoshikawa, K. Takahata, N. Sawano, and Y. Shibata, “Robust and large scale distributed disaster information system over internet and Japan Gigabit Network,” IEEE 21st International Conference on Advanced Information Networking and Applications, pp. 762–768 (2007).

[12] Amazon Web Services (AWS), https://aws.amazon.com/?nc1=h_ls, (2016).

[13] Microsoft Azure, https://azure.microsoft.com/, (2016).

[14] Cloudn, http://www.ntt.com/business/services/cloud/iaas/cloudn.html, (2016).

[15] Zabbix, http://www.zabbix.com/, (2016).

[16] JMeter, http://jmeter.apache.org/, (2016).

[17] Amazon EC2, https://aws.amazon.com/ec2/?nc1=h_ls, (2016).

[18] C.D. Murta and G.N. Dutra, “Modeling HTTP service times,” IEEE Global Telecommunications Conference, GLOBECOM'04, Vol. 2, pp. 972–976 (2004).

[19] UnixBench, https://github.com/kdlucas/byte-unixbench, (2016).

[20] Cassandra, http://cassandra.apache.org/, (2016).

[21] N. Matsuura, M. Ohata, K. Ohta, H. Inamura, T. Mizuno, and H. Mineno, “A Proposal of the Distributed Data Management System for Large-scale Sensor Data,” IPSJ Journal, Vol.54, No.2, pp.721–729 (2013). (in Japanese).

[22] F. Toriumi, K. Shinoda, T. Sakaki, K. Kazama, S. Kurihara, and I. Noda, “Analysis of retweet under the Great East Japan Earthquake,” IPSJ SIG Technical Report, Vol. 2012-ICS-168, No. 3, pp. 1–6 (2012). (in Japanese).

[23] Twitter, https://twitter.com/, (2016).

(Received October 7, 2016)
(Reviced June 16, 2017)

Masaki Nagata
received his M.E. and Ph.D. degrees from Shizuoka University in 2012 and 2017, respectively. He works for the ASP Division of Avance System Corporation in Hamamatsu city, Japan. He is engaged in web system development such as safety confirmation system “ANPIC” and educational ICT system “SACASS”. He also belongs to the Center for Information Infrastructure, Shizuoka University. He is also a member of IPSJ and IEICE.

Yusuke Abe
received his B.S. and M.S. degrees from Shizuoka University in 2007 and 2009, respectively. He works for the ASP Division of Avance System Corporation in Hamamatsu city, Japan. He is engaged in web system development such as safety confirmation system “ANPIC” and educational ICT system “SACASS”.

Misato Fukui
She works for the ASP Division of Avance System Corporation in Hamamatsu city, japan. She is engaged in web system development such as safety confirmation system “ANPI-C” and educational ICT system “SACASS”.

Chihiro Isobe
She works for the ASP Division of Avance System Corporation in Hamamatsu city, japan. She is engaged in web system development such as safety confirmation system “ANPI-C” and educational ICT system “SACASS”.

Hiroshi Mineno
received his B.E. and M.E. degrees from Shizuoka University in 1997 and 1999, respectively. In 2006, he received his Ph.D. degree from the information science and electrical engineering of Kyushu University. Between 1999 and 2002, he was a researcher of the NTT Service Integration Laboratories. Currently, he is an Associate Professor in the Department of Computer Science of Shizuoka University. His research interests include intelligent IoT system as well as heterogeneous network convergence. He is also a member of ACM, IEICE, IPSJ, and the Informatics Society.

Removing Ambiguous Message Exchanges in Designing Sequence Diagrams for Developing Asynchronous Communication Program

Satoshi Harauchi[*], Kozo Okano[**], and Shinpei Ogata[**]

[*]Advanced Technology R&D Center, Mitsubishi Electric Corporation, Japan
[**]Electrical and Computer Engineering, Shinshu University, Japan
[*]Harauchi.Satoshi@bc.MitsubishiElectric.co.jp
[**]{okano, ogata}@cs.shinshu-u.ac.jp

Abstract – Eliminating the reworking of designs is critical for developing software systems. Faults and errors in designs must be extracted so that they do not impair subsequent implementation or test phases. When developing communication programs, faults might linger in the programs. Simply detecting them by reviewing them is difficult, especially when designing complicated and asynchronous communication programs. In this paper, we propose a method that detects faults when designing communication programs by focusing on sequence diagrams that represent message exchanges between lifelines to remove the ambiguity about the order of the exchanges. Our method consists of the following procedures. It generates model descriptions and test expressions from sequence diagrams and executes model checking with them. Then it identifies the location of the information in the diagrams at which errors occur in model checking unless the model descriptions satisfy test expressions. Such notifications enable designers to eliminate inconsistency from their diagrams. This paper describes problems of developing sequence diagrams, our method that solves it, and its implementation with UML 2.0 as well as its evaluation. The evaluation result shows that our method is effective, even though its generation time depends on the complexity of the diagrams.

Keywords: Communication programs, Sequence diagrams, Model checking, Promela, Linear Temporal Logic, UML

1 INTRODUCTION

Removing faults and errors from software systems is critical to develop software with high reliability. Faults can be found not only in software implementation but also in its design. Detecting and removing them is much more important in the design stage than in the implementation stage because removing errors from the design stage generally takes a greater cost and effort because a design must be reworked and modified. Hence, such errors must be extracted so that they do not remain in such subsequent phases as implementations or tests.

Detecting faults is also important for developing communication programs. As communication programs become bigger, their design becomes more complicated. Since detecting them by reviewing their complicated designs is difficult, such designs must be supported to detect faults.

In this paper, we propose a method to detect faults when designing communication programs. Our method focuses on the sequence diagrams that represent asynchronous exchanges of messages between lifelines. The diagrams are complicated when complex communications are being designed. Complicated exchanges of messages frequently cause faults because ambiguity about the order of the exchanges remains in the diagrams. This ambiguity shows that the order for receiving messages could not be determined when several messages are asynchronously transmitted to a specific lifeline. Our method seeks to detect the ambiguity and remove the faults in the diagrams.

Our method consists of the following procedures. First, it generates formal descriptions written in Promela [1] from sequence diagrams. Such components as lifelines and messages described in the diagrams correspond to the Promela elements. Combined fragments, which represent such control structures as alt and loop, are also translated into Promela. Next, with Linear Temporal Logic (LTL), it generates test expressions that are obtained from every message for each lifeline. The generated expressions are used for exhaustively checking the order. Then the method executes model checking with formal descriptions and test expressions. Failing to satisfy the expressions for the formal descriptions suggests the existence of ambiguity related to the order of the messages. Our method finally identifies the position in the diagrams that cause an error in the model checking. Such identification helps designers correct the diagrams and remove the errors. Our method is reapplied from the top of the procedure after error removal, and the designers repeatedly apply it until no more errors occur.

We implement the method as a tool with UML 2.0 and evaluate two aspects. The first aspect focuses on the number of identifications generated by the tool and the time spent on the procedures for various sequence diagrams. The second goes to the diagrams applied to a specific product. The evaluation result shows that the method provides ten candidates for modifying the diagrams and four out of ten candidates are required to correct it, based on interviews with the engineers who worked on the above product.

The remainder of our paper provides detailed analysis of our method. We describe the problems for developing sequence diagrams in Section 2, and our method overcomes them in Section 3. We then describe our method's implementation and evaluation in Section 4.

2 PROBLEMS OF DEVELOPING SEQUENCE DIAGRAMS

Figure 1 shows an example of sequence diagrams. Lifelines A, B, and C asynchronously communicate with each other. The figure is used when designing communication specifications. The design is considered completed after reviewing the diagrams, and then the programs are implemented based on the diagrams.

However, after implementation, a fault might occur in Fig. 1, which shows the sequence diagrams if faults occur. “msg6” is sent from lifelines C to B after “msg3”. Lifeline B receives “msg6” before “msg5”. Nevertheless, “msg6” might reach lifeline B after “msg5,” contrary to the design intention. Lifeline B emits an error due to the specification violation.

Removing faults requires time and effort. The time depends on the causes. The fault shown in Fig. 2 cannot be detected in a unit test, but it can often be detected in integration tests. Accordingly, we must rework the design, implementation, and test. The time to repair faults increases in accordance with the number of faults and the diagram complexity.

The error in Fig. 1 was caused by the ambiguity of the sequence diagrams and indicates that the situation cannot be determined in which “msg6” reaches lifeline B. This paper describes how to remove such ambiguity in designing sequence diagrams.

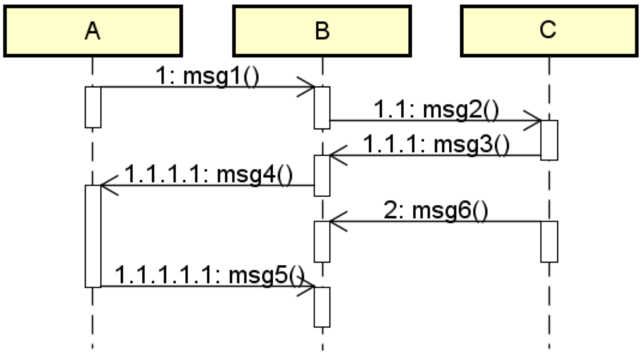


Figure 1: Example of sequence diagrams

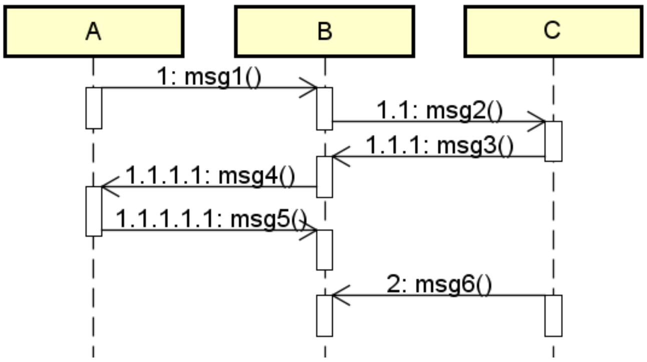


Figure 2: Sequence diagrams in case of a fault

3 PROPOSED METHOD

3.1 Outline

Our method eliminates the ambiguity of the order of messages with semi-automatic modifications of sequence diagrams. Although automatic correction is possible, our procedure provides modification candidates, enables designers to select an appropriate candidate, and corrects the diagrams with the selected candidate.

The input for the method is the diagrams written in XML. The output is the diagrams without ambiguity. The diagram specifications use UML 2.0 [1]. The diagrams allow the asynchronous representation of messages.

Our proposed method consists of these four steps shown in Fig. 3:

STEP 1: Generate formal descriptions;
STEP 2: Generate test expressions;
STEP 3: Perform model checking and generate candidates for modifying diagrams;
STEP 4: Correct the diagrams.

The contribution of this paper is STEP2 and STEP3. In STEP1 we use the existing method [2] which generates formal descriptions. STEP2 provides test expressions used for model checking, and STEP3 indicates the existence of ambiguity and candidates for removing such ambiguity.

We describe the details of each step in the following sections.

3.2 STEP 1 Generate Formal Descriptions

This step generates formal descriptions from the input. An XML is obtained with astah* professional [3]. The formal descriptions are written in Promela [4] and used by the SPIN model checker. Lima’s method [2] generates the descriptions. A lifeline and a message for each execution

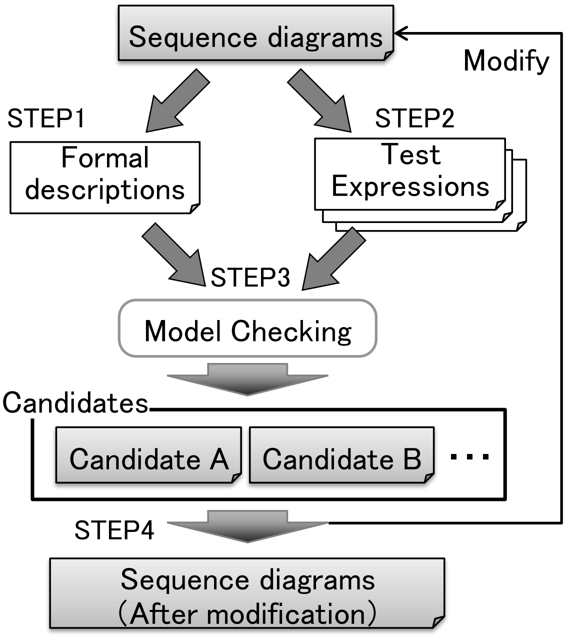


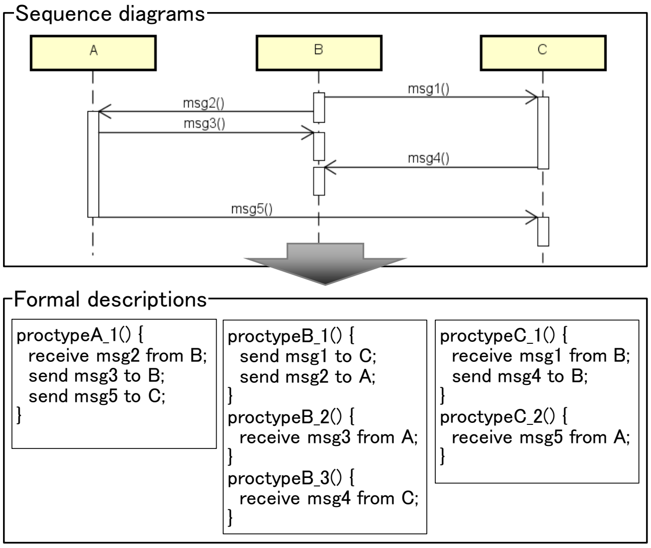
Figure 3: Overview of proposed method

Table 1: Correspondence between sequence diagrams and Promela

Sequence diagrams	Element of Promela	Description in Promela
lifeline	process	proctype {…}
message (label)	message	mtype={m1,…,mn}

message (arrow)	channel	chan chan=[1] of {mtype};
		…
		chan chann=[1] of {mtype};

send event	send	chan!m
receive event	receive	chan?m



```
proctypeA_1() {
  receive msg2 from B;
  send msg3 to B;
  send msg5 to C;
}
```

```
proctypeB_1() {
  send msg1 to C;
  send msg2 to A;
}
proctypeB_2() {
  receive msg3 from A;
}
proctypeB_3() {
  receive msg4 from C;
}
```

```
proctypeC_1() {
  receive msg1 from B;
  send msg4 to B;
}
proctypeC_2() {
  receive msg5 from A;
}
```

Figure 4: Example of generating formal descriptions

specification respectively correspond to a process and a channel with variables defined by Promela. The description in Promela is generated based on the correspondence shown in Table 1.

Figure 4 shows an example of generating formal descriptions. Figure 5 shows a detailed example of the generated descriptions shown in Fig. 4, whose upper part shows sequence diagrams and whose lower part shows a summary of the formal descriptions generated from the diagrams. Process B_1, Process B_2, and Process B_3 are generated since lifeline B has three execution specifications. Each message is translated into two descriptions. For example, "msg1" generates one description by which B sends "msg1" to C and another one by which C receives "msg1" from B.

The generation for each execution specification maintains the order of the messages within the execution specifications. Furthermore, the generated descriptions represent the ambiguity of the order of the asynchronous messages. Lima's method does not generate descriptions for each execution specification.

Combined fragments, which represent control structures such as alt and loop, are converted into Promela as well. We select four fragments, "alt", "par", "loop," and "break," because they are used frequently. The conversion into Promela is shown in Table 2. Description for combined fragment "par" corresponds to the diagrams shown in the right part of Fig. 6.

```
/* Auto Generated Promela File */
/* Message Declaration */
mtype = {msg1, msg2, msg3, msg4, msg5};
/* Channel Declaration */
chan to_A = [20] of {mtype};
chan to_B = [20] of {mtype};
chan to_C = [20] of {mtype};
/* Variable for send and receive */
bool send = false;
bool receive = false;
mtype msg;
/* Process Declaration */
active proctype A_1(){
   d_step{ to_A?msg2; send=false; receive=true;
   msg=msg2;}
   d_step{ send=true; receive=false; msg=msg3;
   to_B!msg3;}
   d_step{ send=true; receive=false; msg=msg5;
   to_C!msg5;} }
active proctype B_1(){
   d_step{ send=true; receive=false; msg=msg1;
   to_C!msg1;}
   d_step{ send=true; receive=false; msg=msg2;
   to_A!msg2;} }
active proctype B_2(){
   …
```

Figure 5: Detailed example of formal descriptions

Table 2: Description in Promela for control structures

Control Structures	Description in Promela
alt, break	if :: (condition 1) -> instruction 1 :: (condition 2) -> instruction 2 … :: (condition n) -> instruction n fi
loop	do :: (condition 1) -> instruction 1 :: (condition 2) -> instruction 2 … :: (condition n) -> instruction n od

par	proctype A() {
	run sub_A()
	AB_msg4?msg4; BSubB?token;}
	proctype B() {
	run sub_B()
	AB_msg4?msg4; BSubB?token;}
	proctype sub_A() {
	atomic{ AB_msg3!msg3; ASubA!token;};}
	proctype sub_B(){
	atomic{ AB_msg3!msg3; ASubA!token;};}

3.3 STEP 2 Generate Test Expressions

This step generates test expressions from the input, written in Linear Temporal Logic (LTL) expressions, which enable the representation of the system states by the changes of time. Time operators are available in addition to the conventional logical operators shown in Table 3. The expressions are used to check whether the diagrams have ambiguity about the order of the messages. Each expression is generated from two messages that are connected.

Figure 4 shows an example. Lifeline B in the sequence diagrams has four message exchanges. The item to be checked is extracted from two adjacent messages, such as "msg2" and "msg1." Since the items in the lifeline are obtained by all of the adjacent messages in relation to lifeline B, they are described as follows:

(a) Whether "msg2" was sent before "msg1" was sent;
(b) Whether "msg3" was received before "msg2" was sent;
(c) Whether "msg4" was received before "msg3" was received;

The method generates the following expressions below from (a) to (c):
(a') (send "msg2") before (send "msg1");
(b') (receive "msg3") before (send "msg2");
(c') (receive "msg4") before (receive "msg3").

The method then translates the above three items into the following test expressions:
(a'') ¬ (send "msg2") ∪(send "msg1");
(b'') ¬ (receive "msg3") ∪ (send "msg2");
(c'') ¬ (receive "msg4") ∪ (receive "msg3").

The method generates test expressions for lifeline A and C in the same way. Two test expression are generated since they have three message exchanges. Consequently, the method generates seven test expressions in all from diagrams shown in Fig. 4.

Table 3: Time operators in LTL

Character	Symbol	Description
Xφ	◦φ	φ will be true in the next state.
Gφ	□φ	φ will always be true after this step.
Fφ	◇φ	φ will be true sometime after this step.
ψUφ	ψ∪φ	φ will be true sometime after this step and ψ will be true until that time.

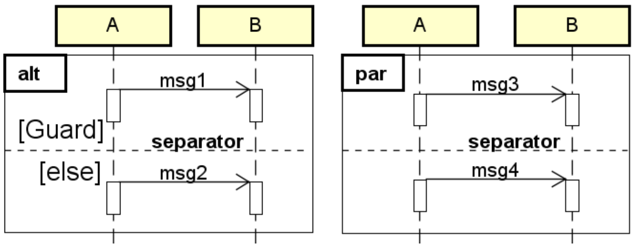


Figure 6: Example of diagrams with "alt" and "par"

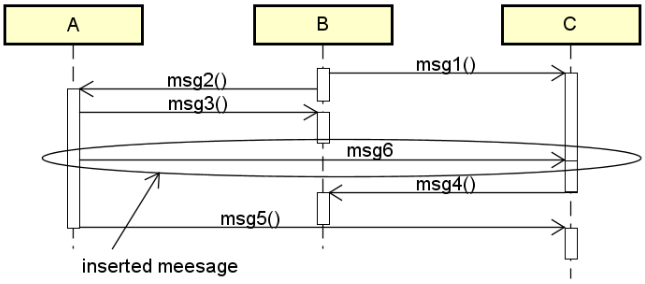


Figure 7: Example of modification

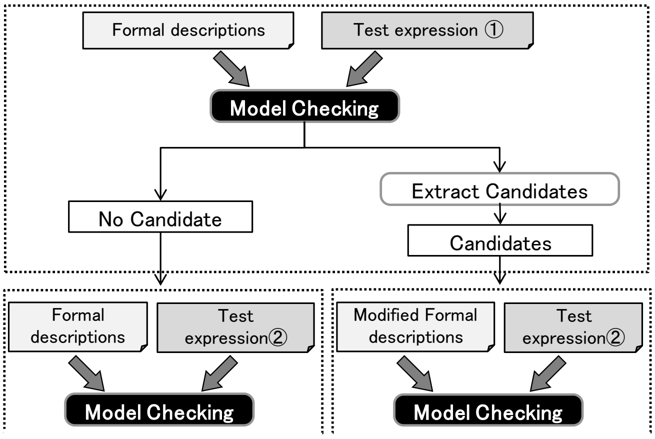


Figure 8: Procedure of generating candidates

Combined fragments are translated into expressions as well. The translation is slightly different from above. For example, "alt" or "par" in Fig. 6 has separators that are divided into operands. "msg1" and "msg2" do not need to be generated since both are executed exclusively. On the other hand, generating expressions for "par" is difficult because "msg3" and "msg4" are executed in parallel. Hence, "alt" or "par" combined fragments are ignored. Only "loop" combined fragment is dealt with for generating expressions.

3.4 STEP 3 Perform Model Checking and Generate Candidates for Modifying Diagrams

This step executes model checking with formal descriptions and test expressions. Then our method provides candidates that indicate how to modify the diagrams. Failing to satisfy the expressions suggests the presence of ambiguity. The failure result gives a pair of two messages described in the test expressions. The candidate shows diagrams with a message inserted between the pair of two messages.

Figure 7 shows an example of the modification candidate. The diagrams shown in Fig. 4 turn out to be ambiguous for the following item.

(c) Whether "msg4" was received before "msg3" was received

Therefore, lifeline A is added to the transmission of "msg6" after sending "msg3," and lifeline C is added to the reception of "msg6" before sending "msg4."

Figure 8 shows the procedure that generates candidates for all of the test expressions. First, our method selects a test expression among all of the expressions and executes model checking with formal descriptions and the selected by the

SPIN model checker. The execution moves to the following processes depending on the checking result.

If no ambiguity exists The method executes model checking with the same formal descriptions and another test expression.

If ambiguity exists The method generates a modification candidate from the test expression. The candidate is presented to a designer, and diagrams are corrected if he decides to apply it. The method then executes model checking with the corrected formal descriptions and another test expression.

The procedure is repeatedly applied and only terminates when model checking is executed for all of the test expressions.

3.5 STEP 4 Correcting the Diagrams

This step corrects the XML with the selected candidate from Section 3.4. The candidate has the information for additional messages, such as the name and the insertion place. The method corrects the definition of the messages and the information related to the lifelines. This step's procedure is completed if all the candidates indicated by the designers are reflected in the diagrams. If all the test expressions pass the model checking after the diagrams are corrected, no existence of ambiguity about the order of messages is proven for the specified diagrams.

4 EVALUATION

We implement our proposed method as a tool with Java and shell scripts to evaluate its performance and the following two aspects:

Aspect 1: Number of modification candidates generated and the time spent on the method's execution

Aspect 2: Candidate evaluation

Aspect 1 focuses on various kinds of sequence diagrams, and Aspect 2 focuses on the diagrams applied to a product. The following are the computer specifications used for the evaluation:

OS: Windows 7 Professional
CPU: Intel Xeon E5607 2.27GHz×2
Memory: 16 GB
SPIN: Version 6.3.2

The size of the state vector for the SPIN model checker was defined as 1024 bytes.

4.1 Evaluation Method

4.1.1 Aspect 1

We collected the sequence diagrams described as examples in existing researches [5]-[9] and applications [10], [11] related to sequence diagrams and produced another sequence diagrams with additional lifelines and messages for specific diagrams of the collected examples. Figure 9 shows the sequence diagrams [9]-1 produced by [9]. The square region in black dotted lines indicates the diagrams [9]. We increased the number of lifelines and messages by extending the original diagrams and applied our tool to them. We enumerated the lifelines, the messages, and the modification candidates and measured the time spent on their execution.

The measurement was executed, assuming that the designers adopted all of the candidates identified by the tool.

4.1.2 Aspect 2

We applied our tool to the diagrams used for the product. The diagrams were rewritten with astah* professional [3]. This aspect checks the ability to detect the faults shown in Fig. 2. We confirmed that the candidates generated by the tool are appropriate to be corrected.

4.2 Evaluation Results

4.2.1 Aspect 1

We applied the tool to eleven sequence diagrams. The evaluation result is shown in Table 4. The eleven diagrams consisted of seven diagrams collected from the references and four where the number of lifelines in reference [9] is increased (described as [9]-1, 2, 3, 4). Columns 1 to 3 show the information in the diagrams and columns 4 to 7 show the result applied to the tool. Columns 1, 2, and 3 respectively indicate the source of the diagrams, the number of lifelines in them, and the number of messages. Column 4 describes the number of candidates generated by the tool. Columns 5 to 7 show the execution time applied to the tool and measured for each step. Column 5 indicates the sum of the time spent on STEPs 1 and 2 since both are executed in parallel with identical input.

No significant differences can be seen in the time spent on STEPs 1 and 2. However, the time of [9]-1, 2, 3 and 4 is large. A large number of lifelines and messages increases the total amount of time. The time spent on STEP 3 becomes large as the number of lifelines or messages in the diagrams increases. The more candidates, the more time will be spent on STEP 4, although no major differences can be observed in the time.

The time spent on STEP 3 in [9]-4 is much smaller than in [9]-3, although the number of lifelines and messages are very large, because the model checking in that case could not be executed due to insufficient memory. Hence, the number of candidates became zero.

We also applied our tool to three more sequence diagrams with combined fragments. The evaluation result is shown in Table 5. The diagrams were extracted from the references. Columns 1 to 6 in Table 5 are the same as in Table 4. Column 7 describes the number of fragments in the diagrams.

The result shows that the time spent on STEPs 1 and 2 was different between the presence and the absence of the combined fragments. This difference reflects the time difference to generate formal descriptions, not test expressions.

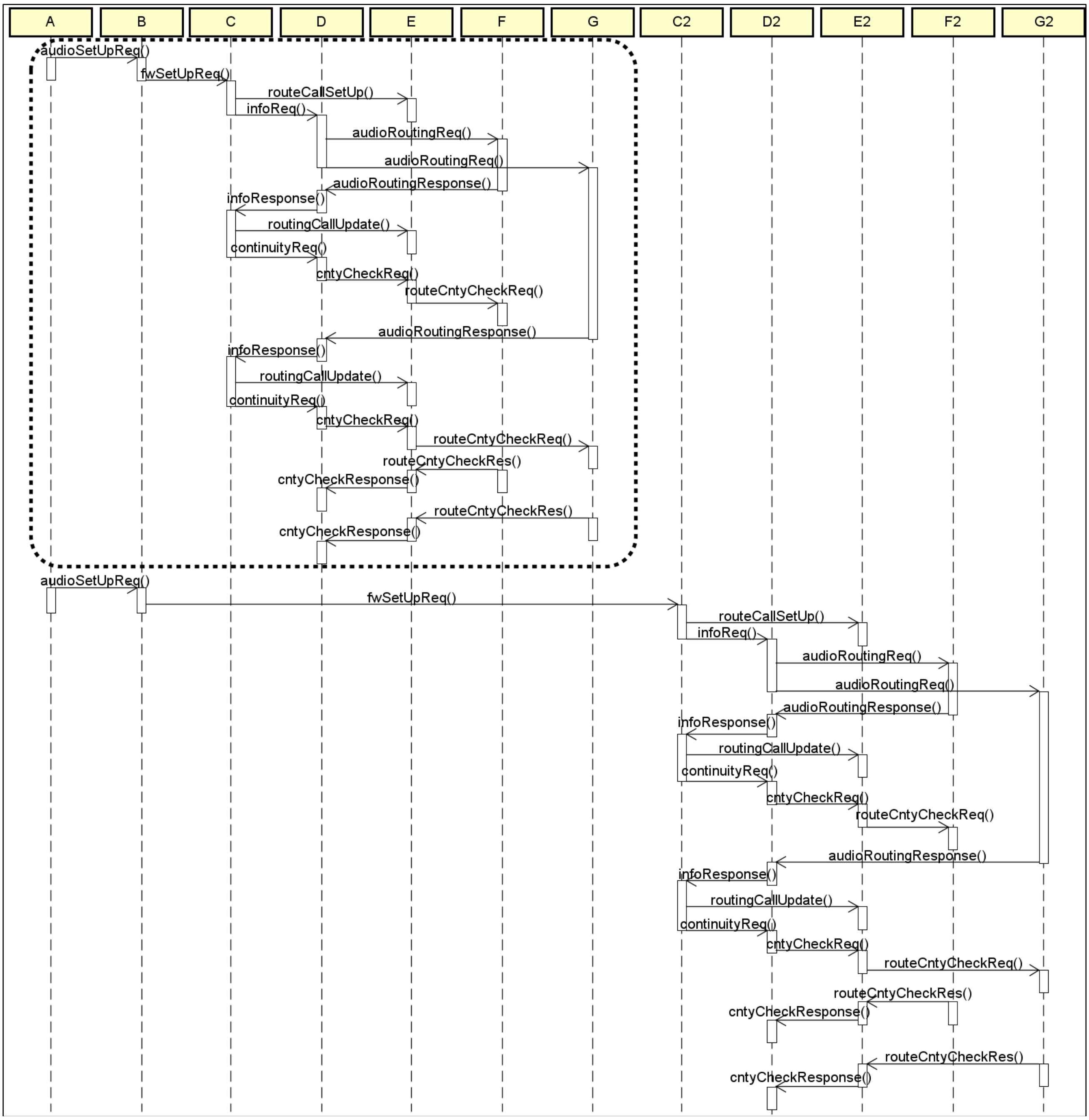


Figure 9: Sequence diagrams [9]-1 produced by [9] for evaluating Aspect 1

Table 4: Evaluation results for Aspect 1

	Lifelines	Messages	Modification candidates	Time spent on execution(seconds)		
				STEP 1,2	STEP 3	STEP 4
[5]	7	11	3	0.44	20.21	0.43
[6]	4	12	2	0.42	25.19	0.41
[7]	3	8	0	0.39	15.55	0.41
[8]	3	4	0	0.37	5.96	0.40
[10]	5	6	1	0.42	8.95	0.41
[11]	6	24	0	0.49	56.15	0.42
[9]	7	22	6	0.46	52.08	0.48
[9]-1	12	44	9	0.58	124.77	0.45
[9]-2	22	88	18	0.98	9136.59	0.52
[9]-3	32	132	24	0.97	14054.17	0.56
[9]-4	37	154	0	1.20	744.73	0.56

Table 5: Evaluation result for Aspect 1 with combined fragments

	Lifelines	Messages	Modification candidates	Time spent on execution(seconds)		Combined fragments
				STEP 1,2	STEP 3	
[12]	4	7	3	1.40	1.3	0
[13]	3	10	4	5.60	16.3	2 (loop, alt)
[6]	4	9	1	5.20	16.0	2 (loop, par)

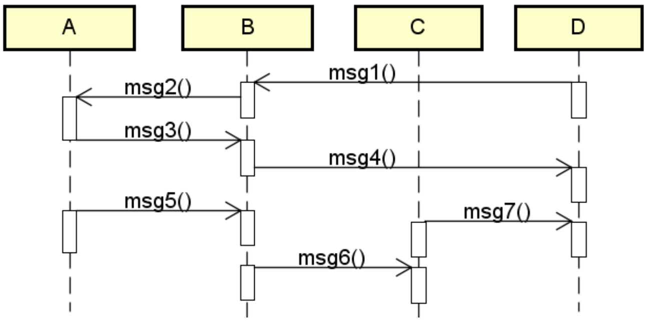


Figure 10: Example of diagrams used for Aspect 2

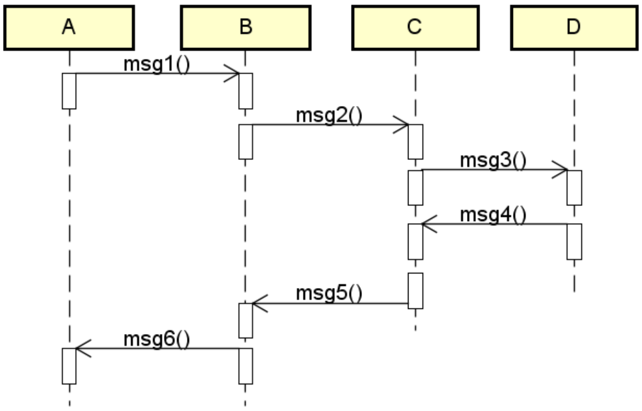


Figure 11: Example of diagrams used for Aspect 2

Table 6: Evaluation result for Aspect 2

	Lifelines	Messages	Modification candidates	Time spent on execution (seconds)		
				STEP 1,2	STEP 3	STEP 4
Figure 10	4	7	5	0.40	14.04	0.62
Figure 11	4	6	5	0.41	12.15	0.52

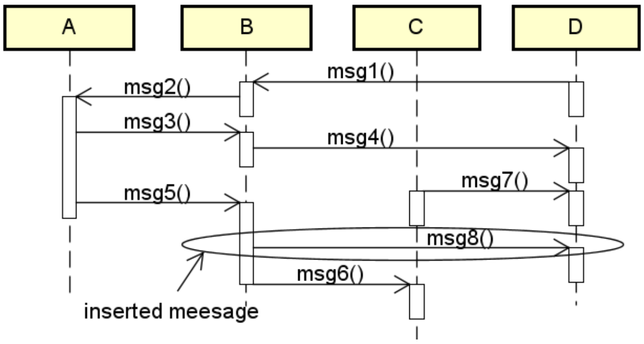


Figure 12: A modification candidate 1

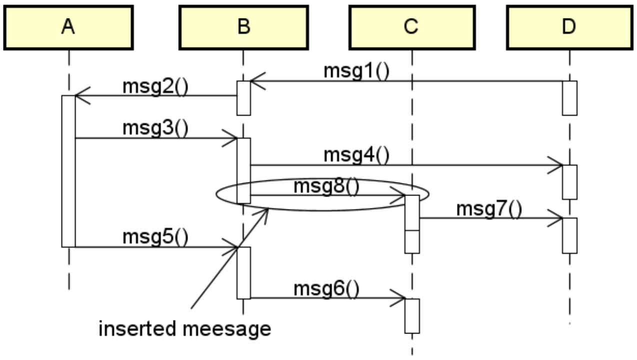


Figure 13: A modification candidate 2

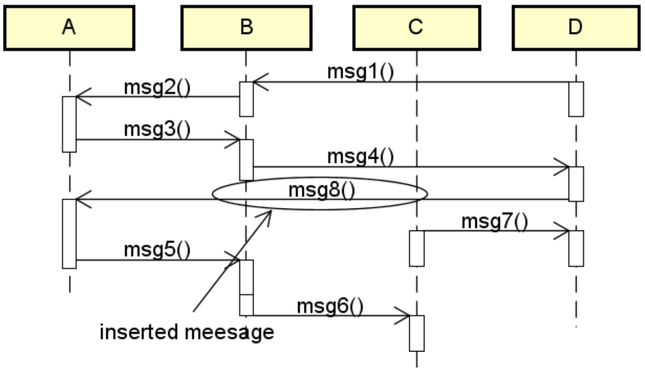


Figure 14: A modification candidate 3

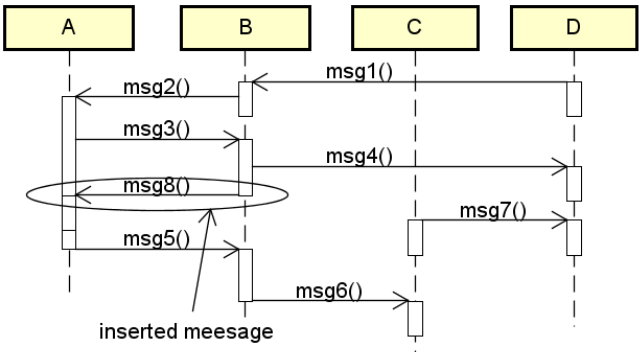


Figure 15: A modification candidate 4

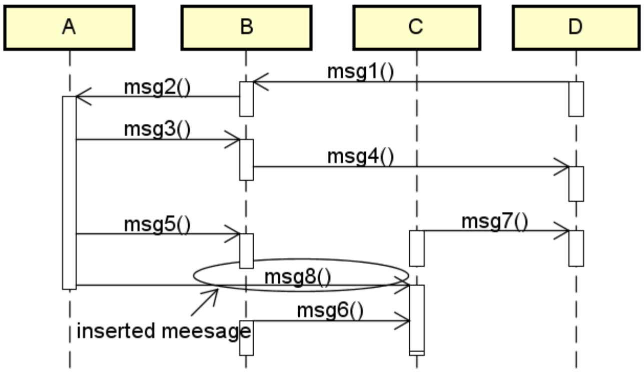


Figure 16: A modification candidate 5

Table 7: Removed ambiguity for each candidate

Candidate	Ambiguity
Figure 12	“msg7” might reach lifeline D after “msg6”
Figure 13	“msg7” might reach lifeline D before “msg4”
Figure 14	“msg5” might reach lifeline B before “msg4”
Figure 15	“msg5” might reach lifeline B before “msg3”
Figure 16	“msg6” might reach lifeline C before “msg7”

4.2.2 Aspect 2

Some of the diagrams used for the product were supplied by factories for evaluating Aspect 2. We selected two diagrams from the supplied diagrams and applied the tool to them where ambiguity about the order of the messages might exist. The applied diagrams are shown in Fig. 10 and Fig. 11. The application result is shown in Table 6 whose columns are the same as in Table 4.

Next we describe all of the candidates obtained by our tool for Fig. 10. The candidates are shown from Fig. 12 to Fig. 16. For instance, Fig. 12 indicates the modification that inserts “msg8”. The candidate is generated from the ambiguity that “msg7” might reach lifeline D after “msg6”.

Table 7 shows the removed ambiguity for all of the candidates. The result shows that our proposed method can detect the ambiguity and generate the candidates for each ambiguity for the sequence diagrams used for the products.

We requested the engineers who developed the product to check the ten candidates including from Fig. 12 to Fig. 16. We confirm their validity by selecting one appropriate answer from the following three options:

(1) This modification must be done.
(2) This modification does not need to be done.
(3) This modification should not be done.

Option (1) was selected for four candidates including Fig. 12. (2) was selected for five candidates and (3) for one candidate.

The answers show that the method detect necessary candidates, but several candidates are of no use. Unnecessary candidates are caused by constraints for modifying sequence diagrams. Figure 10 and Fig. 11 has the lifeline which corresponds to the equipment procured externally. It is impossible to modify sequence diagrams related to such equipment. Hence, the engineers selected answer (2) or (3). We should consider the constraint in order to exclude unnecessary candidates when applying to the design for products.

4.3 Evaluation Validity

In Aspect 1 we collected the sequence diagrams from existing researches and tools and evaluated the number of modification candidates and the time spent on the execution by applying the tool to the diagrams. In this paper we only applied the tool to eleven sequence diagrams. Since the number of lifelines and messages written in the diagrams is limited, we might obtain a different result when applying the tool to large-scaled diagrams.

In Aspect 2 we confirmed the possibility of detecting faults and generated modification candidates from diagrams with ambiguity developed for the product. We only used two diagrams for our evaluations. We must obtain a large variety of diagrams for various products and evaluate them to acquire more general results.

5 RELATED WORKS

Lima et al. proposed a method that generates Promela from sequence diagrams and detects faults with model checking [2]. This method shows representation written in Promela for almost all of the diagrams described in UML 2.0. They implement this method as an eclipse plugin and confirm fault detection by giving appropriate test expressions.

Miyamoto et al. proposed a method that converts the specifications of software written in state diagrams and deployment diagrams in Promela representation [14]. The input for both diagrams is XML produced by astah* professional. Their method generates Promela by translating instances in the deployment diagrams into processes and translating the transitions in state diagrams into processing that executes each process. Converting the patterns of the above specifications in UML into LTL expressions enables the execution of SPIN model checking without describing complicated expressions.

Nagata et al. proposed a method that generates communication programs from the specifications of communication protocols described with sequence diagrams [15]. Their method, which defines the protocols with the diagrams and a format that represents the content of the messages, generates programs from the above definitions. The generation derives exception handling from fault tree diagrams and appends it to the programs in normal processing. The method reduces the overlooking of exception handling required for the occurrence of exceptions for communication programs.

Tiwari et al. proposed a method that generates test cases with activity diagrams that describe software specifications [16]. Their method obtains the conditions under which a system terminates normally from diagrams that represent the processing flow. Their method acquires fault tree diagrams by reversing the conditions and generates test cases where the system terminates both normally and abnormally.

Kaleeswaran et al. proposed a method that detects faults from programs and test suites and shows candidates for correcting the faults [17]. Their method modifies programs based on points specified by toolset Zoltar [18] which automatically localizes faults. Since the modification is then executed by selecting the candidates, it enables semi-automatic corrections.

6 CONCLUSION

This paper proposed a method that detects faults when designing sequence diagrams that describe the asynchronous exchanges of messages. Our method transforms formal descriptions written in Promela and test expressions written in Linear Temporal Logic (LTL) from sequence diagrams and executes model checking for all of the expressions with the descriptions. When an error occurs in an execution, it provides information in diagrams, enabling designers to remove the faults and protect consistency.

We implemented and evaluated our method with two aspects. In the first aspect, we measured the amount of information and the time spent on our method's execution. In the second one, we applied our method to the diagrams used by a product. The application generated ten pieces of information and evaluated their validity. According to interviews with engineers, about 40% of the information is effective for correcting the diagrams. Future work will apply our method to various developments of diagrams and increase the number and the kinds of candidates.

REFERENCES

[1] "UML2.0," http://www.omg.org/spec/UML/2.0/, retrieved on September 13, 2016.

[2] V. Lima, C. Talhi, D. Mouheb, M. Debbabi, L. Wang, and M. Pourzandi, "Formal Verification and Validation of UML 2.0 Sequence Diagrams using Source and Destination of Messages," Electronic Notes in Theoretical Computer Science, vol. 254, pp. 143-160 (2009).

[3] "astah* professional," http://astah.net/editions/ professional, retrieved on September 13, 2016.

[4] G. J. Holzmann, "The model checker SPIN," IEEE Transactions on Software Engineering, vol. 23, no. 5, pp. 279-295 (1997).

[5] P. Baker, P. Bristow, C. Jervis, D. King, R. Thomson, B. Mitchell, and S. Burton, "Detecting and Resolving Semantic Pathologies in UML Sequence Diagrams," Proceedings of the 10th European Software Engineering Conference held jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 50-59 (2005).

[6] S. Bernardi, S. Donatelli, and J. Merseguer, "From UML Sequence Diagrams and Statecharts to Analyzable Petri Net models," Proceedings of the 3rd International Workshop on Software and Performance, pp. 35-45 (2002).

[7] D. Harel and S. Maoz, "Assert and Negate Revisited: Modal Semantics for UML Sequence Diagrams," Software & Systems Modeling, vol. 7, no. 2, pp. 237-252 (2008).

[8] H. Shen, R. Krishnan, R. Slavin, and J. Niu, "Sequence Diagram Aided Privacy Policy Specification," IEEE Transactions on Dependable and Secure Computing, pp. 381-393 (2014).

[9] B. Mitchell, "Characterizing Communication Channel Deadlocks in Sequence Diagrams," IEEE Transactions on Software Engineering, vol. 34, no. 3, pp. 305-320 (2008).

[10] "Lucidchart," https://www.lucidchart.com/, retrieved on September 13, 2016.

[11] "tracemodeler," http://www.tracemodeler.com/, retrieved on September 13, 2016.

[12] H. Shen, R. Krishnan, R. Slavin, and J. Niu, "Sequence Diagram Aided Privacy Policy Specification," IEEE Transactions on Dependable and Secure Computing, no. 99 (2014).

[13] D. Harel and S. Maoz, "Assert and Negate Revisited: Modal Semantics for UML Sequence Diagrams," Software & Systems Modeling, vol. 7, no. 2, pp. 237-252 (2008).

[14] N. Miyamoto and K. Wasaki, "Automatic Conversion from the Specification on UML Description to PROMELA Model for SPIN Model Checker," Forum on Information Technology, vol. 9, no. 1, pp. 311-314 (2010).

[15] T. Nagata, S. Harauchi, M. Kitamura, T. Yamaji, and Y. Ueno, "A Method to Create Network Communication Programs by Deriving Exception Handling from Fault Tree Diagram," Forum on Information Technology, vol. 11, pp. 45-48 (2012).

[16] S. Tiwari and A. Gupta, "An Approach to Generate Safety Validation Test Cases from UML Activity Diagram," Proceedings of the 20th Asia-Pacific Software Engineering Conference, pp. 189-198 (2013).

[17] S. Kaleeswaran, V. Tulsian, A. Kanade, and A. Orso, "MintHint: Automated Synthesis of Repair Hints," Proceedings of the 36th International Conference on Software Engineering, pp. 266-276 (2014).

[18] T. Janssen, R. Abreu, and A. J. C. van Gemund, "Zoltar: A Toolset for Automatic Fault Localization," Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering, pp. 662-664 (2009).

(Received October 7, 2016)
(Reviced May 3, 2017)

Satoshi Harauchi received BE and ME degrees in Information Sciences from Kyoto University in 1996 and 1998, respectively. Since 1998, he has been at the Advanced technology R&D center of Mitsubishi Electric Corporation and is currently interested in software engineering for social infrastructure system. He is a member of IEICE and JSASS.

Kozo Okano received BE, ME, and PhD degrees in Information and Computer Sciences from Osaka University in 1990, 1992, and 1995, respectively. From 2002 to 2015, he was an associate professor at the Graduate School of Information Science and Technology of Osaka University. In 2002 and 2003, he was a visiting researcher at the Department of Computer Science of the University of Kent in Canterbury and a visiting lecturer at the School of Computer Science of the University of Birmingham, respectively. Since 2015, he has been an associate professor at the Department of Computer Science and Engineering, Shinshu University. His current research interests include formal methods for software and information system design. He is a member of IEEE, IEICE, and IPSJ.

Shinpei Ogata is an assistant professor of the Graduate School of Science and Technology in Shinshu University, Japan. He received a PhD from Shibaura Institute of Technology, Japan in 2012. His current research interests include model-driven engineering for information system development. He is a member of IEEE, ACM, IEICE, and IPSJ.

Submission Guidance

About IJIS

International Journal of Informatics Society (ISSN 1883-4566) is published in one volume of three issues a year. One should be a member of Informatics Society for the submission of the article at least. A submission article is reviewed at least two reviewer. The online version of the journal is available at the following site: http://www.infsoc.org.

Aims and Scope of Informatics Society

The evolution of informatics heralds a new information society. It provides more convenience to our life. Informatics and technologies have been integrated by various fields. For example, mathematics, linguistics, logics, engineering, and new fields will join it. Especially, we are continuing to maintain an awareness of informatics and communication convergence. Informatics Society is the organization that tries to develop informatics and technologies with this convergence. International Journal of Informatics Society (IJIS) is the journal of Informatics Society.

Areas of interest include, but are not limited to:

Computer supported cooperative work and groupware
Intelligent transport system
Distributed Computing
Multi-media communication
Information systems
Mobile computing
Ubiquitous computing

Instruction to Authors

For detailed instructions please refer to the Authors Corner on our Web site, http://www.infsoc.org/.

Submission of manuscripts: There is no limitation of page count as full papers, each of which will be subject to a full review process. An electronic, PDF-based submission of papers is mandatory. Download and use the LaTeX2e or Microsoft Word sample IJIS formats.

http://www.infsoc.org/IJIS-Format.pdf

LaTeX2e

LaTeX2e files (ZIP) http://www.infsoc.org/template_IJIS.zip

Microsoft Word^{TM}

Sample document http://www.infsoc.org/sample_IJIS.doc

Please send the PDF file of your paper to secretariat@infsoc.org with the following information:

Title, Author: Name (Affiliation), Name (Affiliation), Corresponding Author. Address, Tel, Fax, E-mail:

Copyright

For all copying, reprint, or republication permission, write to: Copyrights and Permissions Department, Informatics Society, secretariat@infsoc.org.

Publisher

Address: Informatics Laboratory, 3-41 Tsujimachi, Kitaku, Nagoya 462-0032, Japan
E-mail: secretariat@infsoc.org

CONTENTS