The Object-Oriented Database Technology for the Elevator Monitoring System

Kye-Young Lim, Jae-Yong Lee
LG Industrial Systems Co., Ltd.
Building Systems R&D LAB., 6th Street Export IND. CORP.
1385-1 Juan-dong, Nam-gu, Incheon 402-200, Korea

Abstract — Elevator owners increasingly prefer elevators having improved performance and safe operation. This has stimulated the development of elevator monitoring systems. The elevator monitoring systems need various application programs and a lot of information necessarily. The elevator monitoring systems need the database to monitor numerous elevators simultaneously. Since the object-oriented database system has many advantages on managing and maintaining the data structure, we have adopted the object-oriented database for elevator monitoring system. The objectives of this paper are to show the restrictions and the possible problems when using a model of the relational database system for the elevator monitoring system, to show how a model of the relational database system can be extended to a model of the object-oriented database system in order to overcome the restrictions, and to show advantages and future plans when applying the object-oriented database technology to the elevator monitoring system.

1. Introduction

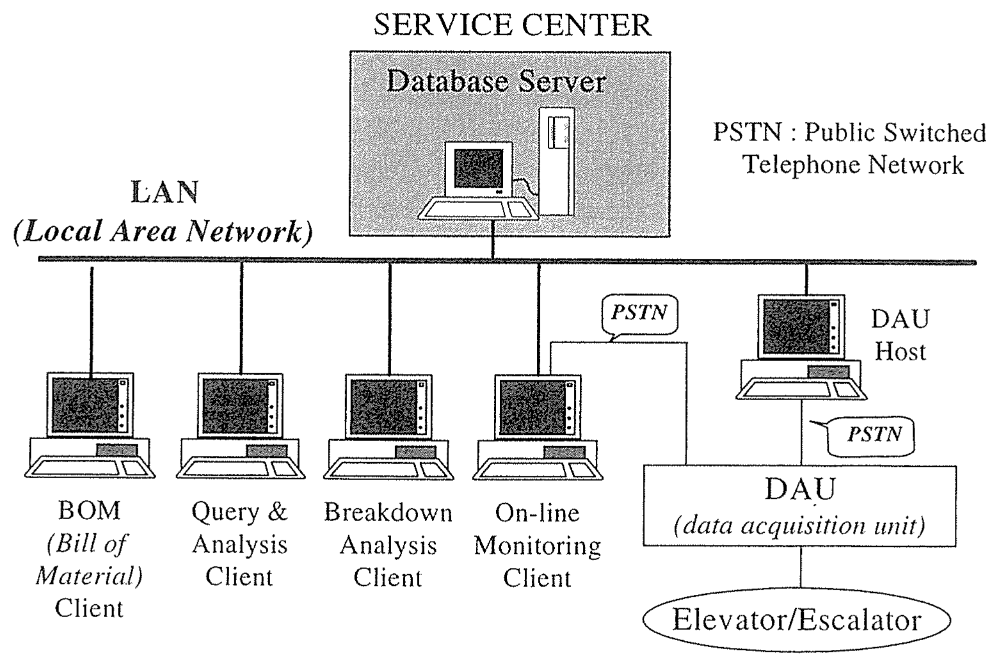

Figure 1. Elevator Remote Monitoring System Structure

As most buildings become higher and bigger in modern society, the demand for more efficient elevators is rapidly increasing. While the number of supply for elevators grows, the number of failure has also been growing. As a result, consumer's dissatisfaction is increasing considerably. A new maintenance system is needed not only to receive product failure promptly to take proper action, but also to perceive failure or its sign in advance to take positive action before a customer files his claim. In addition, an integrated computerized maintenance system is needed to establish to collect objective and precise information on customer and elevator quality and use the collected information organically and operate effective organization. A new elevator remote monitoring system (Figure 1) for basic maintenance has been made to solve the above matter to supervise elevator failure and operation contents, operation environment, and performance, etc, through telephone line 24 hours a day by which the collected data could be analyzed to diagnose failure cause of each elevator or recognize failure occurrence sign to prevent failure in advance.

Remote monitoring system shall be equipped with database server which can store a lot of information from many elevators which need various kinds of applicable program enough to analyze statistics, diagnose failure, evaluate component durability, give work order and inquire information. Therefore database is necessary by all means to store such various kinds of information consistently and the result of various applicable programs easily[1]. We have applied Object Relational Database (ORDB) called UniSQL to manage remote monitoring system information effectively.

This thesis has the following purpose:
Firstly, the thesis presents problems and limitations which may occur when relational database (RDB) is used for remote monitoring system. Secondly, the thesis provides how to overcome the above problems and limitations by using object relational database. In this regard this thesis intends to present how to change general relational schema to object relational schema. Lastly, the thesis presents the advantage and the future plan which may be obtained by applying object-oriented technology to remote monitoring system[2].

2. Relational database limitation

Now relational database has been mainly used to manage general information. Relational database may express data by relational data structure[3]. This thesis explains briefly various limitations which may happen when a relational database is used to handle remote monitoring system information.

1) Elevator remote monitoring system includes a lot of complicated data (component, failure report/action, elevator structure, etc.). When such a complicated data structure is realized by multiple relational tuple, it is difficult to comprehend overall structure. In addition, complicated operation such as join operation must be performed to extract this

kind of data, which takes much time to do.

2) Relational database provides basic built-in data type only such as number, monetary, character, and short symbol, etc. However this kind of built-in data type has difficulty to express the data with composite relations such as component information or elevator structure information. In other words, a variety of user-define data types must be assisted to realize composite relational data easily.

3) Relational database provides stored procedure, which is not related with data and also cannot be reused. However, depending on data feature, remote monitoring system requires various applicable programs that are related to each data. If this one is realized by storage procedure of relational database which cannot be reused, new stored procedure shall be prepared according to structure change, which cannot be also associated with data. In other words, its object encapsulation is difficult. Such matter makes system difficult to be maintained and repaired.

Relational database has many restrictions in realizing effective remote monitoring system owing to the above mentioned problems. Therefore, we have used object relational database of UniSQL and object-oriented technology to expand relational model to object-oriented model. It improved the above problems considerably.

3. UniSQL function expanded more than relational model

UniSQL has expanded relational database function by using object-oriented technology, which has considerably alleviated limitations from relational database[4, 5].

Expansion 1 : Assist user-define data type.

The table, which exists in database, may be used as data type of attribute or row. This can be done by object identity (OID) concepts and all of objects have sole and lasting OID in database. This OID is also used to refer to another tuple, by which multiple relational table's data can be directly approached without join operation. This reduced time considerably to extract complicated data compared with relational database.

Expansion 2 : More than one data value may exist at the intersection of column and row of the table.

More than one value may be given at the intersection of column and row, by which data relation of 'one-to-many' and 'many-to-many' are expressed easily.

Expansion 3 : Table may have its corresponding procedure.

In object-oriented concept, the object consists of the data to express object status and the method to deal with this data, which are generally related to program. With this, we can manage data and its applicable programs in database consistently.

Expansion 4 : Table may be composed of hierarchical structure.

Table may have not only more than one child table but also more than one parent table as well. Child table may not only inherit (reuse) rows and methods of parent table but also redefine rows and methods of parent table.

4. Convert the existing relational schema to object relational schema

Relational database system may be converted to object relational database system by the following steps. Firstly, table must be reconstructed in inheritance structure, which may convert system structure to convenient and simple structure to understand and furthermore give good effect on reducing composition relationship. Secondly, one-to-one, one-to-many relations must be reconstructed by using OID pointer. Thirdly, many-to-many relation must be reconstructed by using OID pointer. Lastly, each table must be connected with its corresponding method. Each step is explained in detail as follows:

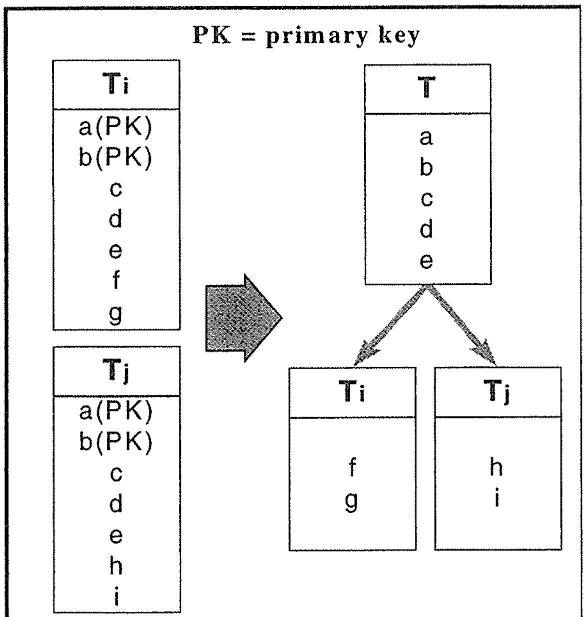

Figure 2.1 Diagram of 1.I

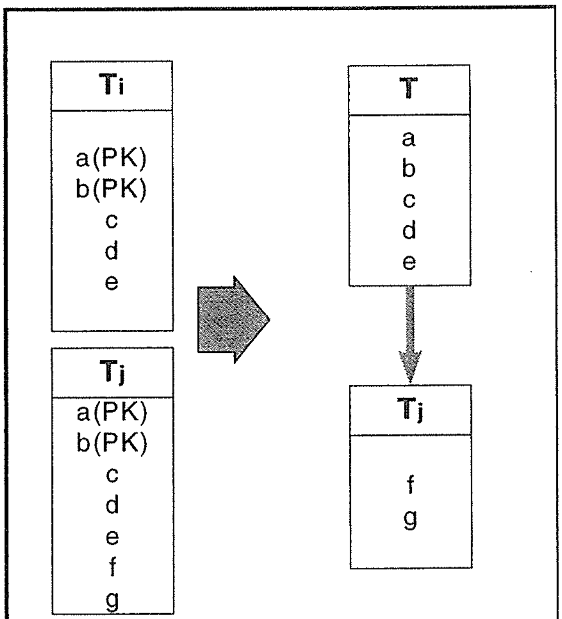

Figure 2.2 Diagram of 1.II

1st step : Table with common attributes must be reconstructed through inheritance again. The following two cases shall be considered:

I. If tables of T_i and T_j have same primary key attributes, new table of T having both primary key shared by T_i and T_j and other common attributes must be constructed and T_i and T_j shall be redefined as subtable of T. These common attributes shall be also removed from T_i and T_j (Figure 2.1).

II. If T_i and T_j have same primary key attributes and T_j includes all the attributes of T_i, T_j removes T_i's common attributes from T_i to reconstruct T_j of T_i's subtable (Figure 2.2).

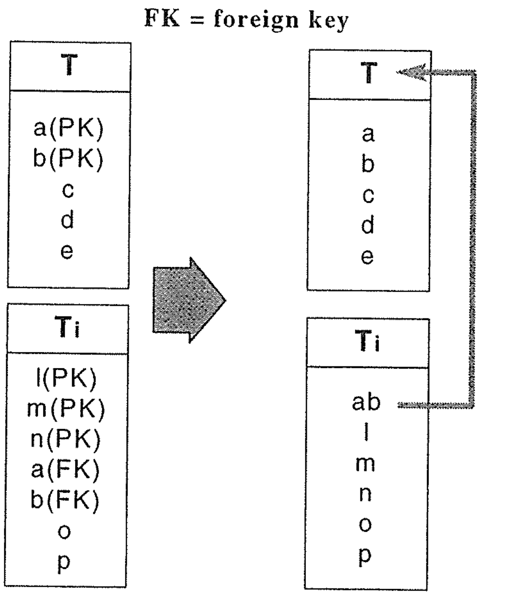

Figure 2.3 Diagram of 2.I

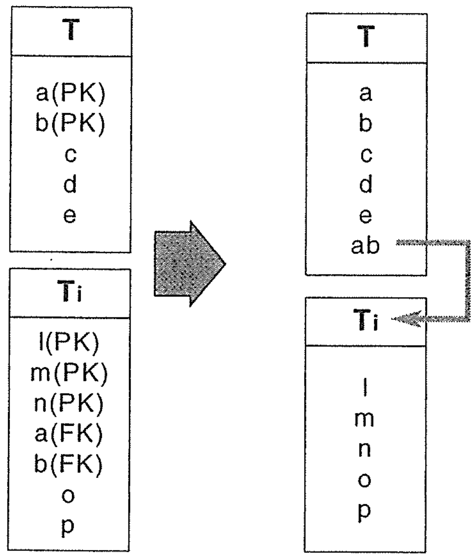

Figure 2.4 Diagram of 2.II

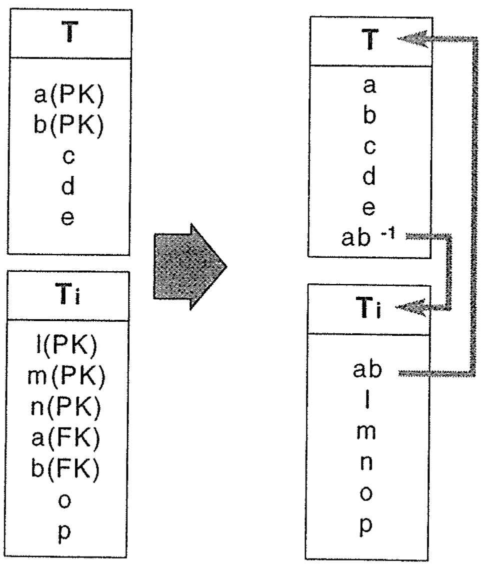

Figure 2.5 Diagram of 2.III

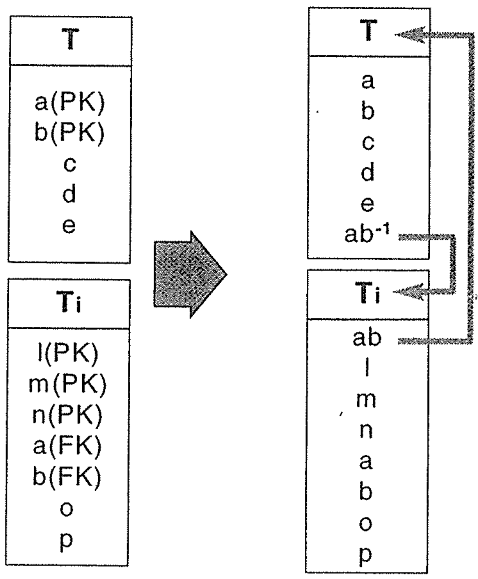

Figure 2.6 Diagram of 2.IV

2nd step : Following step reconstructs relations of primary key/foreign key reference.

Where k_p is primary key of table T and T_i is all the tables having attributes corresponding to k_p, those attributes shall be foreign key. If T_i has foreign key only of T, a relation may exist between both tables, which shall be determined to be 'one-to-one' or 'one-to-many'. Four kinds of methods may be applied to one-to-one relation as follows:

I. The foreign key attributes of T_i shall be replaced with single attribute with T domain (Figure 2.3).

II. New attribute with T_i domain shall be added to T after removing foreign key attributes of T_i (Figure 2.4).

III. The foreign key attributes of T_i shall not only be replaced with single attribute with T domain but also another new attribute with T_i domain shall be added To T. This may establish inverse link between T and T_i (Figure 2.5).

IV. While T_i maintains foreign key, both new attribute with T domain and another new attribute with T_i domain shall be added to T_i, T. This makes inverse link between both tables and also allows to reuse the existing applicable program even by minor revision (Figure 2.6).

The one-to-many relation may be made by revising one-to-one relation among the above four methods as follows: If each tuple of T has 0 or more records, it is necessary to add new attribute with set(T_i) domain to T. If each tuple of T_i has 0 or more records, it is necessary to add new attribute with set (T) domain.

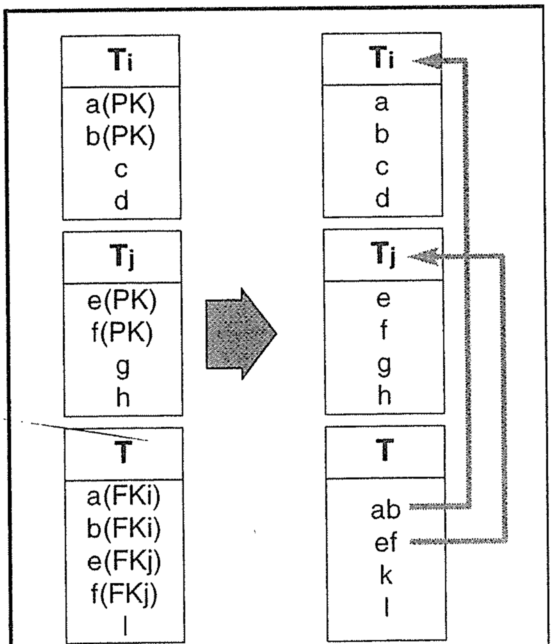

Figure 2.7 Diagram of 3.I

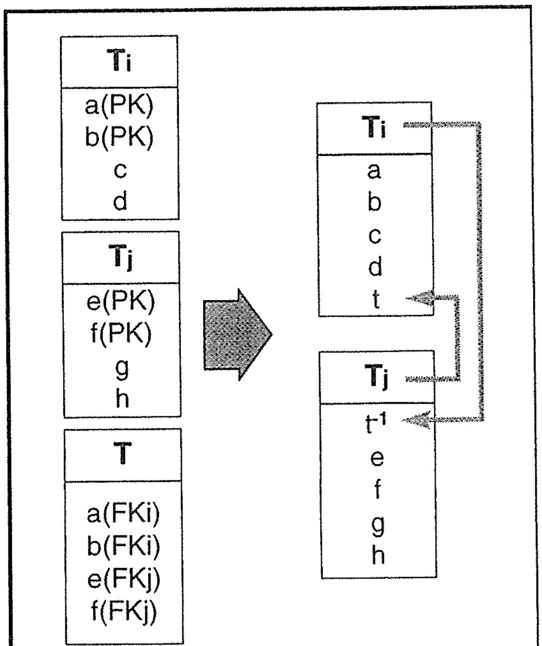

Figure 2.8 Diagram of 3.II

3rd step : 'Many-to-many' relation needs more complicated process. If T is a table having primary key attributes of T_i and T_j, many-to-many relation may exist between T_i and T_j. Two kinds of methods can be applied:

I. Where T has other attributes than foreign key, foreign key of T, which corresponds to T_i, shall be replaced by single attribute with T_i domain, while foreign key of T, which corresponds to T_j, shall be replaced by single attribute with T_j domain (Figure 2.7).

II. Where T has not other attributes than foreign key, the above method may be used. However, it is more useful to remove T and add new attribute with set(T_j) domain, or add new attribute with set(T_i) domain(Figure 2.8).

4th step : The last stage connects methods necessary for various tables. Method is a program to define operation of each tuple of the table so that suitable program may be reused among the existing applicable programs. Also new method may be also needed to establish.

Based on the above methods, the existing relational schema can be converted to object relational schema which is suitable to remote monitoring system. These methods may be revised or supplemented according to the feature of each problems.

5. Experiment result and examples

Remote monitoring system, which has been realized by object relational database system (ORDBs), has advantage over that of relational database system (RDBs) as follows:

① When user-defined data type and OID are used

The query response time of ORDBs, which has been constructed by object relational schema and OID, is much quicker than that of RDBs as shown in the following experiment result. And also total modeling structure may be also understood easily.

- **Relational Schema Query :**

query statement :

SELECT receipt_time, receipt_code, receiver, maintenance_number, customer_no, car_no FROM receipt_class, elevator_class
WHERE maintenance_no = maintenance_number;

average response time : <u>*28 ~ 30 secs*</u>

- **Object Relational Schema Query :**

query statement :

SELECT receipt_time, receipt_code, receiver, elevator.maintenance_number, elevator.customer_no, elevator.car_no FROM receipt_class;

average response time : <u>*10 ~ 13 secs*</u>

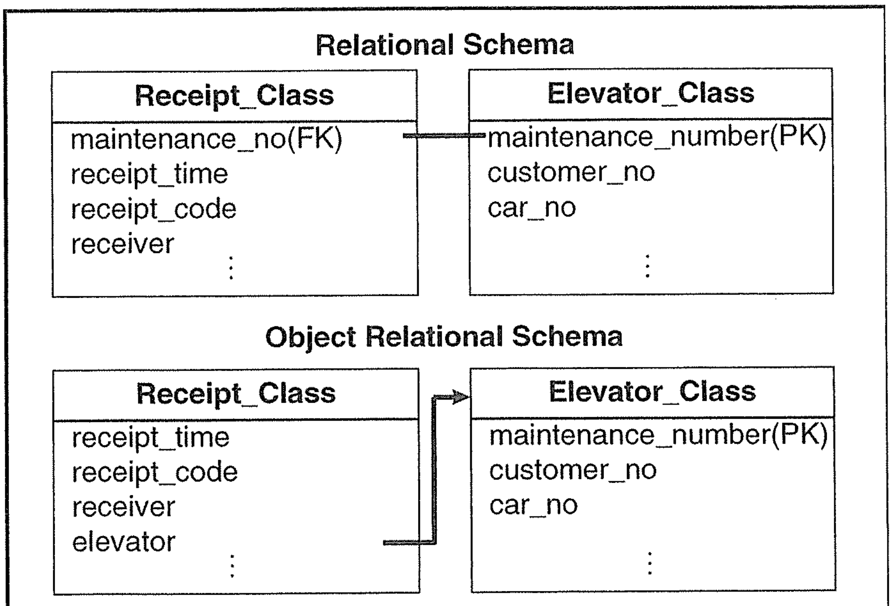

Figure 3.1 Schema for Comparison of Response Time

Join operation and OID pointer method have difference of select response time two times as shown in the above. In addition complicate structural query has difference more than about 10 times.

② The example of both inheritance structure and composition structure using user-define data type

Figure 3.2 and Figure 3.3 are examples of inheritance and composition structure.

Modeling of class or object, which is based on object-oriented concept by using inheritance and composition, may express entity of the actual situation as it is, which is easier to understand overall structure and more convenient to do modeling than expressing matters by flat table at RDBs. Data overlap of no use may be also solved by using the existing schema again.

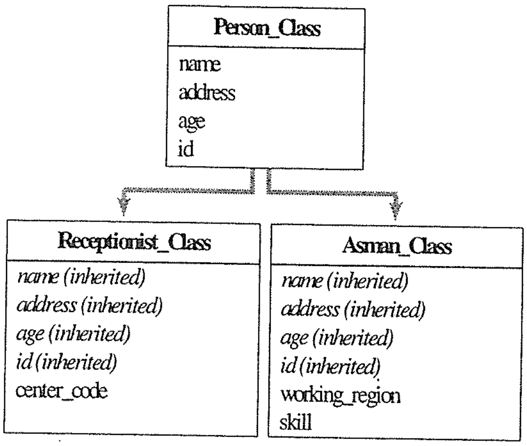

Figure 3.2 An example of inheritance

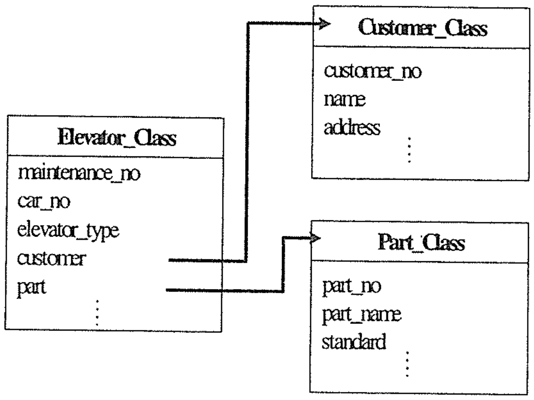

Figure 3.3 An example of composition

③ **The example of encapsulation**

Table may be added by its data and related method as shown in Figure 3.4. Using this method allows corresponding function to work. Whenever new data is added, auto trouble alarm method of the example gives alarm sound. Frequently used or special function may be managed together with data to be simply manipulated by method change when a function is needed to revise.

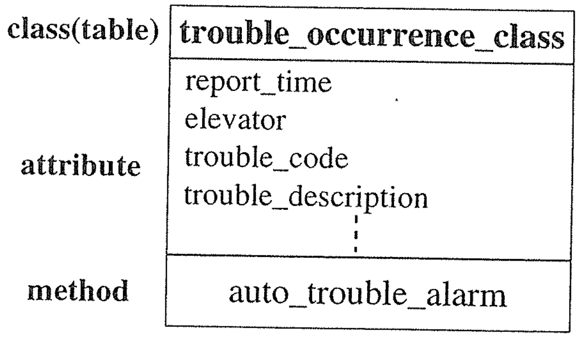

Figure 3.4 An example of encapsulation

6. Future development

In addition to the above, ORDBs has a lot of advantages. ORDBs is able to manipulate multimedia data easily so that photo information, drawing information, noise and other multimedia data can be stored to apply to the applicable program in future[6].

Common gateway interface (CGI) information itself will be studied to store in ORDBs in order to perform the method, which has been designated to each client, through Web browser.

7. Conclusions

This thesis describes the advantage when elevator remote monitoring system adopts the database, which has introduced object-oriented concept, to manage information. Object relational database may overcome the occurred problems considerably. We can effectively manage information of system which monitors many elevators and establish database structure which may take immediate action upon continuous request of users.

8. About the authors

Dr. K.Y.Lim obtained BSc from Seoul National University, MSc and Ph.D from State University of New York at Stony Brook in Electrical Engineering. After graduation he joined Research and Development Laboratory of LG Industrial System (LGIS) where he participated in and led the development of controllers for robotics, machining tools and elevator systems. Now, he is a Executive Director in the Building Systems R&D LAB. of LGIS. Dr. Lim has a special interest in Elevator, Escalator and Parking system.

Mr. J.Y.Lee obtained BSc and MSc from Seoul National University in Control Engineering. After graduation he joined LG Industrial System (LGIS) where he participated in Elevator Remote Monitoring System project. Now, he is a Researcher in the Building Systems R&D LAB. of LGIS.

9. References

[1] James Rumbaugh, et al, "Object-Oriented Modeling and Design", Prentice-Hall, Inc., pp 366-368, 1991

[2] A. Snyder, "Encapsulation and Inheritance in Object-Oriented Programming Languages", Proceedings of the First ACM Conference on Object-Oriented Programming Systems, Languages and Applications, SIGPLAN Notices, vol. 21, no. 11, pp 38-45, 1986

[3] C. J. Date, "An Introduction to Database Systems Volume II", Addison Wesley, pp 188-192, 1986

[4] Grady Booch, el al, "Object-Oriented Analysis and Design with Applications", The Benjamin Cummings Publishing Company, Inc., 1994

[5] Won Kim, Frederick H. Lochovsky, "Object-Oriented Concepts, Databases, and Applications", Addisson Wesley, 1989

[6] "UniSQL/X User's manual", UniSQL, Inc., 1995