Data Warehousing Massive Real-time Elevator Signals and Maintenance Records

Yi-Yang Yang[1], Yain-Whar Si[2], and Wai-Leong Leong[3]

Faculty of Science and Technology

University of Macau

Abstract

Elevators are regarded as important modern day necessities for urban communities. Although elevators are widely used in various installations, the actual control and monitoring of these systems is given less attention by the research community. Specifically, the problem of maintaining and analyzing massive real-time elevator control signals has been largely ignored by data mining community. Every elevator consists of at least one onboard controller (circuit board). Like any other real-time systems, an onboard elevator controller can generate a large volume of signals. The size of real-time signals and maintenance records generated annually from a number of high-rise housing estate elevators can reach hundreds of Gigabytes. In this paper, we describe a data warehousing approach for managing massive real-time elevator signals. Our prototype system shows significant reduction in storage requirement and allows efficient query processing across massive real-time data.

I. INTRODUCTION

Elevators are widely used in modern society. Elevators can be found in virtually everywhere from multi-story buildings to underground mines. Although they are extensively used in various forms of installations and being investigated from different perspectives (e.g. acceleration [9], destination control [10], etc), less attention has been devoted to the managing of real-time data from elevator control and monitoring systems. Specifically, the problem of maintaining and analyzing massive real-time elevator control signals has been largely ignored by data mining community.

An onboard elevator controller usually generates huge amount of data during day-to-day operation. The generated information is usually processed by a monitoring system in real-time. The vast amount of information includes:

- *Normal Data:* This is used to indicate the information such as current location of the elevator, speed, moving direction, and so on.
- *Signals:* which are used to indicate any faults detected within the system. For instance, overheating of the engine, error in closing/opening the doors, error in stopping at designated floor levels, etc.
- *Events:* which are used to activate maintenance actions for engineers. Events are generated when a particular type of signal is accumulated over time. Specifically, events are used to indicate that there could be serious hardware/software problems in the system and urgent maintenance may be required.

In addition, elevators are required to be serviced periodically by maintenance staff. In certain circumstances, immediate repairing may be needed when events are detected in the monitoring system. These maintenance operations may also generate a large number of records depending on the number of elevators being maintained. To the best of authors' knowledge, majority of today elevator control systems [1,2,3] are designed with relational databases. However, such approach does not scale when a large number of elevators are usually required to be constantly monitored since the amount of data generated by such monitoring and control systems can overwhelm the capacity of a typical database system within a short time.

Moreover, accumulated signals and maintenance records are recognized as valuable sources of information for future analysis by engineers. The analysis can reveal information such as frequency of usage, rate of faults detected, and other vital statistics. Data mining such information is less effective when traditional databases are used for storing of massive elevator signals and maintenance records. To alleviate these problems, we design an elevator control and

[1] *yyygou@gmail.com*
[2] *fstasp@umac.mo*
[3] *marklwl_black@hotmail.com*

978-1-4244-1706-3/08/$25.00 ©2008 IEEE.

monitoring system which is capable of managing massive real-time signals and maintenance records using Data Warehousing techniques. A Data Warehouse is "a subject oriented, integrated, nonvolatile, and time variant collection of data in support of management's decision" [7]. Data warehouses are mainly used by managers and decision makers to extract information quickly and easily in order to answer questions about their business. One of the techniques in designing data warehouse is dimensional modeling [8]. By using this technique, storage requirement can be reduced without incurring significant degradation on query performance while retaining the capability for future data analysis.

The design of an elevator control and monitoring system is described in section II. Data generated by an onboard elevator controller is discussed in section III. The detail of maintenance records is described in section IV. The design of the data warehouse is illustrated in section V. Potential queries for analyzing massive real-time data are described in section VI. In section VII we briefly review related work before summarizing our ideas in section VIII.

II. ELEVATOR CONTROL AND MONITORING SYSTEM

The architecture of an elevator control and monitoring system is depicted in Figure 1.

The system consists of a number of onboard elevator controllers, a central server, a monitoring and control terminal, a back-office maintenance and scheduling system, a database server, and a data warehouse server. Onboard elevator controllers are connected to the central server via a public/private communication network. For the sake of simplicity, we have omitted a standby central server.

In Figure 1, the Monitoring and Control Terminal is designed to send instructions to the onboard elevators controllers via the central server. These instructions may include directing an elevator to move to a designated floor or directing an elevator to stop at a certain position. Likewise, real-time status information of elevators is pulled from the central server and forwarded to the monitoring and control terminal.

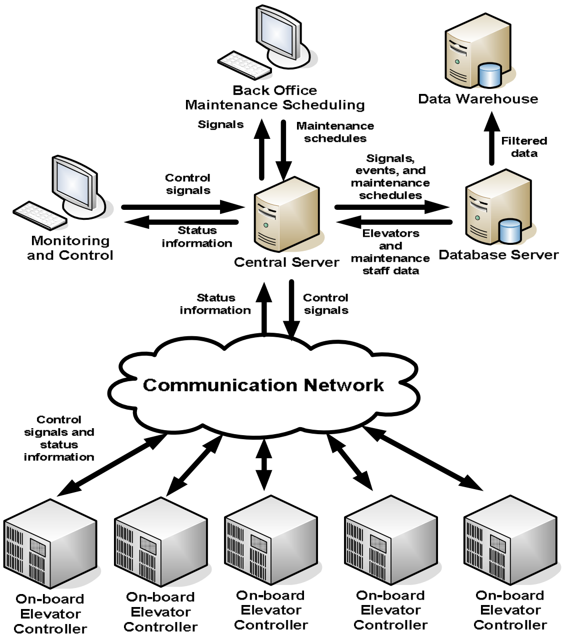


Figure 1. Elevator control system

A database system is used to store details of maintenance staff, their availability, and recent signals and events received from onboard controllers. A back-office maintenance scheduling system regularly retrieves these data from the database server to generate a maintenance schedule on hourly or daily basic. Signals and maintenance records accumulate in the database server over time and once it reaches a predefined threshold, the data from the database server is extracted, transformed, and achieved into the data warehouse server.

III. ELEVATOR STATUS

On-board elevator controllers usually generate a huge amount of data in real-time, which includes Normal Data, Signals, and Events representing every aspects of the elevator status.

Normal Data: Normal data is used for monitoring elevator status and is required to be sent in a fixed time interval. Normal data contains detailed information of the elevator such as: speed, motor temperature, pressed button in each floor and so on. They are formatted as a constant data stream since their generation and sending rates are predefined by the onboard elevator controller and synchronized with the central server. One of the main constraints of the Normal data is that the information contained has to be analyzed by the central server in real-time. The description of normal data is described in Table I.

TABLE I
NORMAL DATA

No	Normal Data Field	Description
1	Elevator Status	The elevator running mode such as Normal, Idle, Fire Mode…
2	Current Floor	The current floor of the elevator
3	Moving direction	/
4	Door State	Close or open
5	Internal Call	Call Button panel state in lift for each floor (0- the floor is not called, 1-the floor is called internally)
6	External Call	Call Button panel state in each floor (0-press neither of button, 1-press up button, 2 press down button, 3-press both buttons)
7	Floor Status	Able or disable to arrive at certain floor (0-disable, 1-able)
8	Running Speed	/
9	Lift Temperature	/
10	Lift Engine Temperature	/

The size of fields 5, 6, and 7 from Table I is variable since it relates to the number of floors in the installation. For instance, if we assume that a building has three floors and the elevator is able to reach at all of them, then the *Floor Status* would be 111. If two passengers get into the lift in 1st floor and if one of them wishes to go to the 2nd floor and the other wishes to go to the 3rd floor, then the *Internal Call* field would be represented as 011. At the same time, if there is a user waiting for the lift in the 1st floor presses the up button and some users in 2nd floor press both up and down buttons, then the current *External Call* field would be represented as 130. The central server extracts the Normal data from the data packets and forwards it to the Monitoring and Control Terminal. The Normal Data is then used for the real-time display of the current status of the elevator as well as for the animation.

Signals: Signals are usually used to indicate minor system faults. Signals are also accumulated over time to generate Events which may require urgent attention. For instance, overload problem occurs frequently in certain elevators during rush hours. As a result, Central Server will receive a large number of data packets which include the *Overload Signal*. In normal situation, such signals are not critical and may not require immediate maintenance action. However, persistent presence of *Overload Signal* may indicate a serious problem in the elevator operation and thus lead to an event which requires a maintenance action.

Signals are also used in recording emergency call from elevator internal telephone system. Emergency calls can be categorized into user related and elevator related. User related emergency calls are generated whenever a passenger uses the internal telephone system to seek help from security staff for illness or criminal activities. When this happens, it generates an Emergency Call Signal.

Elevator related emergency calls are generated when technical problems are detected in the elevator systems. For example, a passenger may press the Emergency Call Button when he/she is trapped in the elevator due to malfunctioned doors. In such cases, the onboard elevator controller transmits the corresponding signals which in turn trigger an event at the Central Server for maintenance action. Signals are also a valuable source of information for both elevator retailers and maintenance engineers. Based on historical signals and normal data, data mining and user behavior analysis can be performed. The descriptions of some of the sample signals are described in Table II.

TABLE II
SIGNALS

Signal ID	Signal	Description
1	Lift Start	Lift start
2	Monitoring Signal	Controller starts monitoring
6	Door Fault	Door cannot close or open
7	Motoring Fault	Cannot monitoring
8	Safety Circuit Fault	Safety Circuit has fault
9	Rapid Stop Fault	Lift cannot Rapid Stop
11	Emergency Call Button Pressed	There is a emergency call from the lift
12	Intercom	User is communicating with security staff by using intercom
14	Cage Arrival Signal	Lift Cage arrive at certain position
15	Full Signal	Lift is full of load
16	Overload Signal	Lift is overload

Events: Elevator Events are used to notify the central server that there are some faults in the elevator and may need a maintenance activity. However, complexity of maintenance actions can vary from one situation to another. In some cases, detected faults can only be solved by an experienced engineer whereas in some situations, a simple system reset by a technician can be sufficient. Due to the space limitation, we have chosen some of the sample events for illustration in Table III.

TABLE III
EVENTS

Event ID	Abbreviated Error Message	Description	Action	May Cause Emergency Call	Repair Time (min)
1	Door or locksw. Shaft	Lift door or lock switch shaft release	Check door lock at cl.12, cl.12a, cl.13[4].	No	65
2	excess pressure	Engine excess pressure	Reset Lift	No	1
3	malfu. brake running	Elevator brakes when it is running, be locked soon	Check the transducer	Yes	50
4	malfu. close brake	Elevator doesn't brake when it should stop, be locked soon	Check the transducer	Yes	50

The *Error Messages* in second column in Table III are abbreviated since the display panels at the sites of elevator installation have limited capacity. The *Description* field denotes the full explanation of the Event. The *Action* field in fourth column suggests possible maintenance actions to the current event. *Repair Time* field is used to indicate the estimated duration for that maintenance action. Depending on the seriousness of an event, an engineer or a technician may be dispatched to the site.

IV. Maintenance Records

Elevator systems are regularly serviced by the maintenance personals to guarantee a smooth, safe, and uninterrupted operation. Maintenance records are crucial for both *Elevator Suppliers* and the *Customer Companies* (the authority of the site where the elevators are installed) for several reasons. For suppliers, maintenance records are useful in determining:

- the type of elevators which require frequent servicing,
- future maintenance schedule and job allocations,
- parts which are needed to be repaired regularly,
- parts which are going to be replaced,
- parts which have been replaced,
- accumulated cost in repair time, and
- accumulated cost in spare parts.

Maintenance records may also be used to monitor the efficiency of a particular staff in servicing activities. For elevator retailers, maintenance records are useful in determining current operational status of the elevators, their safety records, and accumulated downtime during servicing period. Maintenance actions can be performed by engineers and technicians from the elevator company or the security staff from the installation where elevators are located. Examples of maintenance records are shown in Table IV.

TABLE IV
EXAMPLES OF MAINTENANCE RECORDS

ID	Start Date/Time	End Date/Time	Elevator ID	Staff	Action taken	Est. Cost
1	2006/07/13 9:40 am	2006 07/13 10:35 am	1	E	Replace Component	$200
2	2006/03/09 3:40 pm	2006/0 3/09 4:10 pm	2	T	Elevator Lubrication	$50
3	2006/06/09 2:10 pm	2006/0 6/09 2:20 pm	3	S	Elevator Resetting	Free

Legend:
E : Maintenance engineer S: Security staff
T: Technician

V. Data Warehouse Design

Recall that a database is used in our system to store the detail of maintenance personnel and their availability, signals, and events received from the onboard controllers from the elevators currently monitored. When the records stored in the database server reach a predefined threshold, the data is extracted, transformed, and achieved into the data warehouse server. Such measures have significant impact on the system performance.

- By storing real-time data in the database, the system is guaranteed to produce a fast processing time (response time) for the control and monitoring tasks.
- By storing historical data in the data warehouse, the system allows efficient query execution and analysis on the massive elevator signals and maintenance data.

In Table V, we outline the size of a data file transmitted from an Onboard Elevator Controller to the Central Server.

TABLE V
Elevator Real-time State

[4] C1.12, C1.12a and C1.C13a are the ID of particular components in an elevator.

	Title	Type	Size (Byte)
1	Lift_State_ID	Integer	4
2	Lift_ID	Varchar(16)	16
3	State_Date	Date	4
4	State_Time	Time	8
5	Lift_Event	Varchar(256)	256
6	Lift_Signal	Varchar(256)	256
7	Lift_Status	Smallint	2
8	Lift_Current_Floor	Smallint	2
9	Lift_Moving_Direction	Smallint	2
10	Lift_Door_State	Smallint	2
11	Lift_Internal_Call	Varchar(66)	66
12	Lift_External_Call	Varchar(66)	66
13	Lift_Floor_State	Varchar(66)	66
14	Lift_Running_Speed	Real	4
15	Lift_Engine_Temperature	Real	4

The transmitted data file is then stored in a database at the Central Server. In Table V, Lift_Event field and Lift_Signal field are designed as Varchar data types since new events and signals may be added in future expansion of the system[5]. According to this design, if the first event is detected, then Lift_Event field will have the value 1 followed by 255 zeros. The same representation is used for Lift_Signal field. The corresponding database schema for Table V is depicted in Figure 2.

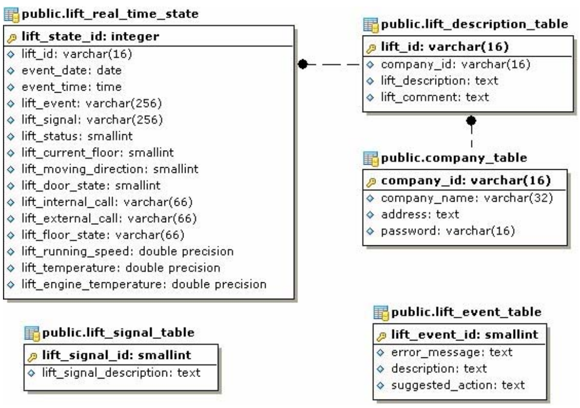


Figure 2. The Database Schema

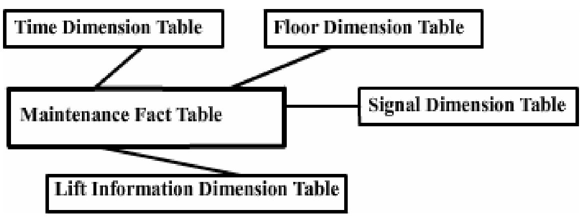
Suppose that 15 Onboard Elevators Controllers (equivalent to the number of elevators within a typical housing estate in Hong Kong and Macau) are connected to the Central Server and each Onboard Controller is configured to send a data file every 5 seconds. Suppose that we store the received signals in a database and in average, we need to save three records (758 * 3 = 2274 Bytes) per second. After one year, the size of the stored records will be 71.71 Gigabytes. In general, a typical elevator company provides maintenance service to more than one housing estate and the number of elevators being monitored can easily reach to hundreds. In such situation, deploying a traditional database management system becomes highly inefficient since massive amount of data has to be stored and evaluated. Specifically, we identity potential bottlenecks as follow:

- Storage requirement: The system needs to manage massive data sets for future analysis.
- Query execution: Accessing data from event logs is inefficient. Since *Lift_State_ID* is the only index and each record is a snapshot of the elevator current state, scanning the whole data is necessary even for a simple query. Any query on information about floor and signals is also unnecessarily complex since string parsing is required during execution.

Against this background, we strive to achieve two main objectives during the design: to improve efficiency and to reduce storage requirement.

Improving access efficiency: Our first approach is to design a star schema which can be used to store relevant signals and maintenance records in a data warehouse[6] (DW). A star schema typically consists of a 'fact table' in the middle of a schema diagram and surrounded by a set of 'dimension tables'. Fact tables are structured to define business metrics or measurement. They are used to store granular data over time at lowest level of the dimension hierarchies. Dimension tables are structured to define business dimensions and they are collections of related values used as indexes for accessing the data in the fact tables.

A star schema design for the data warehouse model is depicted in Figure 3. We use maintenance information as a fact table, surrounded by four dimension tables: time dimension, lift information dimension, floor dimension and signal dimension. Through primary keys of these tables, we can access historical data from different "point of views". Moreover, with the star schema design, string parsing is done only once when real-time signals are converted into the data warehouse.



[5] At present, there are approximately 100 signals and events defined in our system.

[6] PostgreSQL, http://www.postgresql.org/

Figure 3. Star Schema design for integrating with a Data Warehouse

Reducing storage requirement: In Figure 3, time factor is considered as one of the dimensions. However, an average On-board Elevator Controller sends signals every 5 seconds and according to above star-schema design, time dimension table has to store approximately 365 * 24 * 60 * 60 / 5 = 6307200 records per year and the maintenance fact table has to store approximately 1598496768000 records (by taking into account all dimension tables). Therefore, a typical star schema design still requires a large storage requirement.

To alleviate this problem, we design a novel scheme for reducing storage requirement by combining the fact table and the time dimension table from Figure 4. From our analysis, we find that four time points are needed to be recorded:

- P1: The last maintenance end time,
- P2: The time when the Central Server receive an event which triggers the next maintenance action (Casual Event),
- P3: Maintenance Start Time, and
- P4: Maintenance End Time.

We also find that Event Date and Event Time from Table V can be replaced with a single data type *Timestamp* which occupies less storage space. In addition, we find that certain timestamps (8 Bytes per each timestamp) can be transformed into time durations (4 Bytes per duration) which require less space. For example, in Figure 4, we can replace P1, P2, P3, and P4 with P1, D1, D2, and D3. Therefore, instead of keeping four timestamps we can now store one timestamp and three time durations in each record within the maintenance fact table.

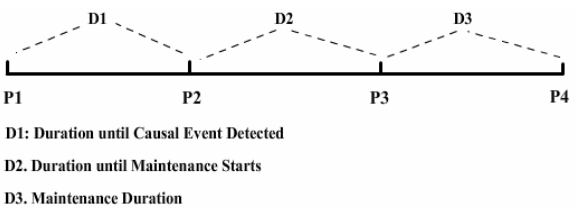


Figure 4. Time information in maintenance fact table

The revised star schema after combining the fact table and the time dimension table is depicted in Figure 5.

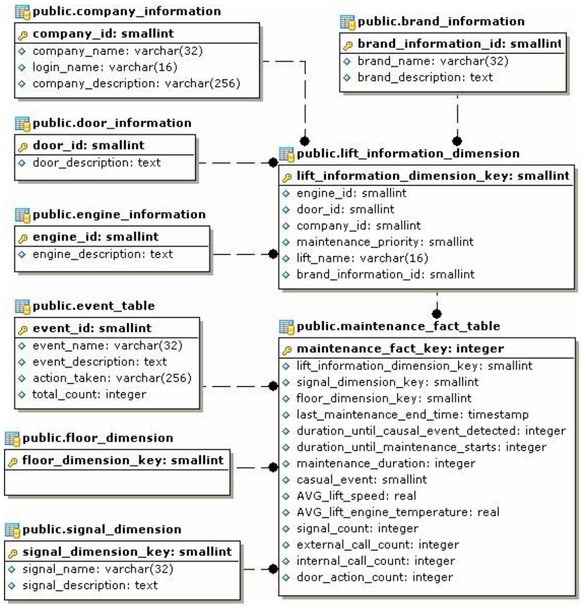


Figure 5. Extended star schema design for data warehouse

In Figure 5, the maintenance fact table is connected to three dimension tables: signal dimension, floor dimension, and lift information dimension. Based on the extended star schema, elevator signals are processed and stored based on interval rather than time instance. In Figure 5, four counters (signal count, external & internal calls count, and door action count) in the maintenance fact table are used to keep track the status of the elevator. These counters are useful when we determine the number of times the elevator door open/close at a particular floor. In addition, further compression of signals is achieved by recording only casual events which trigger maintenance action.

Based on the revised star schema design, we are now able to calculate the size of the data which is going to be stored in the data warehouse for a period of one year for 15 elevators. Note that the *event table* is not a dimension table since not all events can trigger a maintenance action (i.e. for every event, there may not be one corresponding maintenance fact record). The revised content of the maintenance fact table is shown in TABLE VI.

TABLE VI
STORAGE REQUIREMENTS FOR STAR SCHEMA

	Title	Type	Size Bytes
	Maintenance_fact_table		
1	Maintenance_fact_key	Integer	4
2	Lift_information_dimension_key	Smallint	2

3	Signal_dimension_key	Smallint	2
4	Floor_dimension_key	Smallint	2
5	Last_maintenance_end_time	Timestamp	8
6	Duration_until_causal_event_detected	Integer	4
7	AVG_lift_speed	Real	4
8	AVG_lift_engine_temperature	Real	4
9	Signal_count	Integer	4
10	External_call_count	Integer	4
11	Internal_call_count	Integer	4
12	Door_action_count	Integer	4
13	Causal_event	Smallint	2
14	Duration_until_maintenance_start	Integer	4
15	Maintenance_duration	Integer	4
We assume that in the worst case, each lift needs 1 maintenance activities per day, Size = 56 (Size of record) * 15 (records in lift information dimension)* 256 (records in signal dimension)* 66 (records in floor dimension) * 365[7] = 5365324800 Bytes = 5.18 Gigabytes			

With above star schema design, we are able to reduce the storage requirement of signals for 15 elevators over the period of one year from 71.71 Gigabytes to 5.18 Gigabytes (92.78% reduction in size).

VI. QUERY PROCESSING

Based on the extended star schema in Figure 5, engineers are now able to issue queries which can reveal information about the safety of the elevators and the effectiveness of the maintenance actions. Engineers may issue queries such as:

- Find events which are detected most frequently.
```
SELECT * FROM event_table ORDER BY total_count
DESC
```

- List those elevators which generate highest number of events.
```
SELECT lift_information_dimension_key,
COUNT(lift_information_dimension_key) AS lift_times
FROM maintenance_fact_table m
GROUP BY m.lift_information_dimension_key
ORDER BY lift_times DESC
```

- Determine the brand type of lift which generates highest number of events.
```
SELECT brand_information_id,
COUNT(m.lift_information_dimension_key) AS
lift_count
FROM maintenance_fact_table m,
lift_information_dimension l
WHERE l.lift_information_dimension_key =
m.lift_information_dimension_key
GROUP BY l.brand_information_id
ORDER BY lift_count DESC
```

[7] It is also possible to create a period dimension in the star schema.

- Determine the most frequently requested (by pressing internal lift buttons) floor number by passengers.
```
SELECT f.floor_dimension_key,
SUM(m.internal_call_count) AS
floor_internal_call_count
FROM floor_dimension f, maintenance_fact_table m
WHERE f.floor_dimension_key =
m.floor_dimension_key
GROUP BY f.floor_dimension_key
ORDER BY floor_internal_call_count DESC
```

- Average time for servicing a particular event.
```
SELECT causal_event,  avg(maintenance_duration) as
avg_duration
FROM maintenance_fact_table m, event_table e
WHERE m.causal_event = e.event_id
AND e.event_name = 'Error 1'
GROUP BY (causal_event)
```

The implemented Data Warehouse not only enables such queries across massive elevator signals but also increases the query execution time.

VII. RELATED WORK

To the best of our knowledge, there is no reported work on data warehousing real-time elevator signals. However, we have found a number of reports on data warehousing signals from real-time systems. We also find that most of the recent work on data warehousing signals from real-time systems focuses on the design of the real-time system rather than on the data warehousing aspects.

In Zero-Latency Event Sensing and Responding Architecture (ZELESSA) system [6], Tho Manh Nguyen et al. build an infrastructure that can meet the requirements of real-time business system with zero-latency. ZELESSA deals with the problem of many information systems: how to handle the gap between the time when the real-time data is created (received) and the time the related analysis information is becoming available. ZELESSA system maintains two databases: real-time database and a data warehouse.

T. Chi et al. [5] have designed an Ocean Stereo Monitoring System for real-time marine three-dimensional inspection. Deploying such system allows precautions to be taken against any marine calamity. Their approach adopts a star schema design which divides data into five themes: storm tide calamity, red tide calamity etc. The center of the star schema is a fact table, which is surrounded by five different themes tables with “make-up” dimensions such as time dimension, space dimension, sea wave inspection dimension and so on. These five themes make up the center of a normal star schema. However, the actual compression rate of the data is not reported in [5].

In grid structure based data classification [4], Kiran Chelluri et al. consider the problem of very large data warehousing (VLDWH). In [4], data reduction is performed from two perspectives. First, according to time perspective, historical data is divided into several periods such as real-time, post-decade and so on. According to data perspective, the data is further divided based on content. As a result, historical data is transformed into different classes (grids) by taking into account both time and content factors. These classes are assigned with different priorities and the system refines/summarizes those classes with lower priorities to reduce the storage requirement.

H. Gonzalez et al. [11] propose several approaches to compress the data in Radio Frequency Identification (RFID) applications for tracking objects. First, a number of objects are grouped and each group is assigned with common attributes. As a result, a single record can be used to represent a large number of similar objects. Next, RFID data is further reduced and compressed by ignoring some over detailed information. For instance, information of object movement is only retained in granularity of hour. To achieve further compression, object movements are assigned with different priorities and the data on the movements of objects with low priority are summarized or ignored.

VIII. CONCLUSION

In this paper, we describe a data warehousing approach for managing massive elevator signals. We classify real-time elevator data into two categories based on two access modes: real-time access mode and data warehousing access mode. In real-time access mode, we deploy a relational database system to store information generated during the elevator operation for status monitoring with near zero latency performance. In data warehousing access mode, elevator signals are extracted and transformed into a compact star schema based data model. Based on the model, maintenance engineers may issue queries across massive historical data which is generated over years of maintenance activities. Our main contributions in this work are (a) detailed analysis of real-time elevator signals, (b) conceptualization of a star schema for transforming massive real-time data from traditional databases into multi-dimensional databases, and (c) a novel approach to capture time dimension in the data warehouse design. The proposed data warehouse design will be used in future commercial release of the elevator control system. For the future work, we are currently extending the system to allow scheduling of maintenance activities based on evolutionary computing techniques.

ACKNOWLEDGMENT

Authors thank Dr. Vai Man I from University of Macau and Mr. Tong Fai Meng from Macau Hualong Fabrica De Elevadores Lda. for their insightful comments on elevator control system design.

IX. REFERENCES

[1] WINMOS®300, http://www.winmos.de/index_en.htm, Accessed 4 October 2007.

[2] OTIS Elevator Co., http://www.otis.com/, Accessed 4 October 2007.

[3] HITACHI GLOBAL: Elevators, Escalators, and Building Systems, http://www.hitachi.co.jp/Prod/elv/en/index.html, Accessed 4 October 2007.

[4] K. Chelluri, V. Kumar. 'Data Classification and Management in Very Large Data Warehouses', Proceedings of the Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS '01), page 52-57, 2001. IEEE.

[5] T. Chi, X. Zhang, H. Chen, Q. Wang, C. Chen, Y. Lu. 'Study on Ocean Stereo Monitoring Information Data'. Proceedings of Geoscience and Remote Sensing Symposium, 2004, IGARSS '04, pages 2182-2185, 2004. IEEE.

[6] T. M. Nguyen, J. Schiefer, A. M. Tjoa. 'ZELESSA: an enabler for real-time sensing, analyzing and acting on continuous event streams'. International Journal of Business Intelligence and Data Mining, 2(1): page105-141, 2007.

[7] W. Inmon. Building the Data Warehouse, John Wiley and Sons, 1993.

[8] P. Ponniah. Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals, Wiley-Interscience, 2001.

[9] S. Valiviita, S.J. Ovaska, 'Delayless acceleration measurement method for elevator control', IEEE Transactions on Industrial Electronics, Vol. 45, no. 2, page 364-366, Apr. 1998.

[10] J. Koehler and D. Ottiger, 'An AI-based approach to destination control in elevators', AI Magazine. Vol 23, Fall 3, page 59-78, 2002.

[11] H. Gonzalez, J. Han, X. Li and D. Klabjan, 'Warehousing and Analyzing Massive RFID Data Sets', Proceedings of the 22nd International Conference on Data Engineering (ICDE'06). Vol 00, page 83, 2006.