Event-driven Elevator Testing, Control, Monitoring, and Maintenance

Yain-Whar Si[1], Yi-Yang Yang[2], Wai-Leong Leong[3], Sio-Meng Leong, and Chi-Iong Wong
Faculty of Science and Technology
University of Macau
e-mail: fstasp@umac.mo[1], yyygou@gmail.com[2], marklwl_black@hotmail.com[3]

Abstract—**Elevators are considered as important transportation systems for urban communities. Elevators are installed with onboard controllers (circuit boards) and these controllers can generate a large volume of signals and events. In this paper, we describe an event-driven system to test, control, and monitor a large number of on board elevator controllers. The integrated system consists of a virtual controller, control and monitoring terminals, a central server, a playback function with animation, a genetic algorithm based maintenance scheduling module, and a data warehouse for managing massive real-time elevator signals. Based on the event-driven architecture, the proposed system is capable of facilitating faster deployment of new types of elevators. The system also provides engineers with playback functions for troubleshooting any hardware or software errors. In order to reduce overhead cost, the proposed system is designed to optimize resource allocation in maintenance scheduling. By deploying data warehouse technology, the proposed system allows significant reduction of storage requirement for managing real-time signals.**

I. INTRODUCTION

Elevators are extensively used in various forms of installations. Onboard controllers installed within the elevators 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 within an elevator control system includes normal data, signals, events, and maintenance records. Although elevator related issues are investigated from various perspectives (e.g. acceleration [9], destination control [10], group control [21,22,23] etc), less attention has been devoted to the realization of functional components which is capable of utilizing these generated information. Specifically, less research work has been devoted to the properties such as:

1) testing and deployment of new types of elevators,
2) troubleshooting from recorded signals,
3) resource optimization in maintenance scheduling based on real-time events, and
4) managing massive real-time control signals.

In this paper, we describe the development of an event-driven elevator control and monitoring system which aims to achieve above properties. In the proposed system, all crucial activities are triggered by the events (messages) which are either originated from on-board controllers or generated from control and monitoring terminals.

A brief review of related work is given in section II. The design of an elevator control and monitoring system is described in section III. Data involved in overall system is discussed in section IV. The design of the Virtual Controller is given in section V. The Control and Monitoring Terminal is discussed in section VI. The design of the Playback Module is described in section VII. The genetic algorithm based Maintenance Scheduling Module is discussed in section VIII. The star schema design of the Data Warehouse is illustrated in section IX. Section X summarizes our ideas and future work.

II. RELATED WORK

Elevator control and monitoring systems are extensively researched in past decades. In this section, we review related work from four perspectives.

Testing and deployment of new types of elevators: Both hardware and software simulation are extensively used in testing of elevator systems. In hardware-based approaches, Ryu et al. [4] have proposed a dynamic load simulator using electrical inertia. The proposed system can be used to simulate the load characteristics of an elevator system by controlling the torque of DC machine which is much smaller, lighter, and more flexible compared to the conventional one. In [5], Monti et al. have proposed a Real-Time Power Hardware in the Loop (PHIL) platform to emulate the behavior of an electrical machine in order to perform verification tests on power electronic converters. The platform is used to perform quality assurance tests on three-phase power converters by replacing the electromechanical load connected to their terminals. In software-based approaches, simulation has long been considered a tool for testing new elevator dispatching strategy [11], and for designing elevator system [14]. Recently, Cortés et al. [6] have proposed a planning and simulating tool for dynamic vertical traffic. The tool can be used in the planning and design stage of the elevator system for the selection of the type of elevator (number, type of dynamic, capacity, etc.). In elevator traffic simulation [15], Marja-Liisa Siikonen has investigated elevator dynamics and control, call allocation and passenger traffic modeling in a generic building. Although elevator simulation programs are extensively used in educational and commercial environments for testing certain aspects of the control system, less attention has been devoted to emulating both on-board elevator controllers and internal/external floor calls. In this paper, we describe how a virtual controller can be used to emulate such situations.

Troubleshooting from recorded signals: In average, 30 deaths are reported each year in United States from elevator/escalator

1-4244-2384-2/08/$20.00 ©

related incidents [18]. Although a number of studies have been made on elevator electrical design [16], braking [17], and remote video surveillance [19], reconstructing operation sequence from recorded signals has received less attention in elevator control research community. In our approach, a playback module was designed to assist engineers in troubleshooting elevators.

Resource optimization in maintenance scheduling based on real-time events: Although resource allocation/optimization in scheduling is a well known problem, less attention has been devoted to the application of these techniques to elevator maintenance scheduling based on real-time events. In recent work on quality engineering [20], Yamashina et al. have proposed a cost-optimized after-sales strategy for the maintenance of an elevator part. They have derived the replacement probability and the inspection probability of the part using three maintenance methods: Time Based Maintenance (TBM), Condition Based Maintenance (CBM), and Breakdown Maintenance (BM). In our approach, an evolutionary programming was used for optimizing resource allocation in scheduling elevator maintenance tasks.

Managing massive real-time control signals: To the best of our knowledge, there is no reported work on managing large scale real-time elevator signals. However, we have found a number of reports on data warehousing signals from real-time systems. In Zero-Latency Event Sensing and Responding Architecture (ZELESSA) system [26], 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. In [25], Chi et al. have designed an Ocean Stereo Monitoring System for real-time marine three-dimensional inspection. Their approach adopts a star schema design which divides data into five themes: storm tide calamity, red tide calamity etc. In grid structure based data classification [24], Kiran Chelluri et al. consider the problem of very large data warehousing (VLDWH). In [24], 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. In this paper, a data warehousing approach is proposed for managing massive elevator signals. In our approach, elevator signals are extracted from a relational database and transformed into a compact star schema based data model.

III. ELEVATOR CONTROL AND MONITORING SYSTEM

The architecture of an event-driven elevator control and monitoring system is depicted in **Figure 1**.

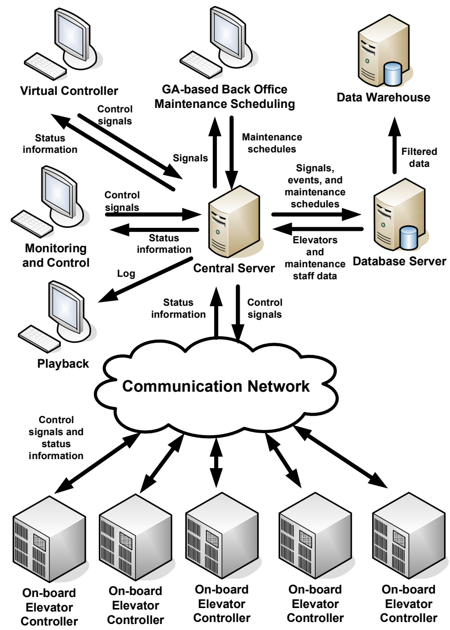

Figure 1. Elevator control and monitoring system

The system consists of a number of onboard elevator controllers, a central server, a virtual controller (for testing), a monitoring and control terminal, genetic algorithm based 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. 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. A virtual controller is used for testing overall performance of the system and for evaluating functions within monitoring and control terminal. The data generated from such testing can also be used for calibrating back office maintenance scheduling module and database related functions. A database system is used to store details of maintenance staff, their availability, and recent signals and events received from onboard controllers. A genetic algorithm based 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.

IV. ELEVATOR DATA

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. Normal data include elevator mode, current floor number, door state, internal call button panel state, external call button panel state, floor status, moving direction, elevator speed, elevator temperature, and engine temperature. The central server extracts the Normal data from the data packets and forwards them to the Monitoring and Control Terminal.

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. Elevator signals include monitoring signal, door fault, motoring fault, safety circuit fault, rapid stop fault, intercom status, cage arrival signal, full load signal, overload signal, and emergency call button signal.

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 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 I.

The *Error Messages* in second column in Table I 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.

TABLE I 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[1].	No	65
2	excess pressure	Engine excess pressure	Reset Lift	No	1
3	malfu. brake running	Elevator brakes when it is running	Check the transducer	Yes	50

***Maintenance records*:** Elevator systems are regularly serviced by the maintenance personals to guarantee a smooth, safe, and uninterrupted operation. Examples of maintenance records are shown in Table II.

TABLE II EXAMPLES OF MAINTENANCE RECORDS

ID	**Start Date/Time**	**End Date/Time**	**Elevator ID**	**Staff**	**Action taken**	**Est. Cost**
1	2006/07/13 9:40	2006/07/13 9:40	1	E	Replace Component	$200
2	2006/03/09 15:40	2006/03/09 15:40	2	T	Elevator Lubrication	$50
3	2006/06/09 14:10	2006/06/09 14:10	3	S	Elevator Resetting	Free

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

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, planning future maintenance schedule and job allocations, identifying parts which are needed to be repaired or replaced regularly, estimating accumulated cost in repair time, and calculating 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.

V. VIRTUAL CONTROLLER

Recent commercial offering of Virtual Elevator Designer [13] allows users to select and visualize the style and material for cab interior design. The virtual controller (see Figure 2) developed in our system allow faster testing, deployment, and fine-tuning of on board elevator controllers.

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

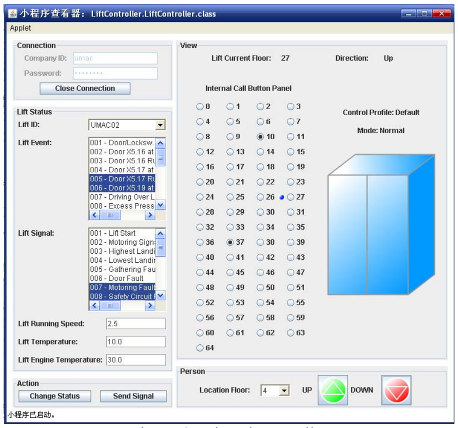

Figure 2. Virtual controller

The virtual controller can be used to generate simulated signals and events without having to install physical onboard elevator controllers. Users can use a virtual controller to simulate any elevator activities and at the same time can observe the real-time status changes of the simulated elevators in an interactive way. In addition to generating signals, events, internal calls, external calls, the virtual controller can be used to generate status information on speed, cage temperature, and engine temperature of the elevator. The simulated activities of elevators from the virtual controller can be monitored through control and monitoring module (see next section). In addition, the data generated is then used to test the performance of real-time database. Due to these capabilities, the virtual controller shortens the time required for testing, installation, and adjusting new types of on board elevator controllers since any changes in signals and events can be quickly tested and adopted within the system.

VI. CONTROL AND MONITORING

Controlling and monitoring are the core functions of the proposed event-based framework. The Control Terminal (see Figure 3) 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. Other control actions include setting the modes of an elevator (e.g. normal, fire, etc.), setting the door state (e.g. open or close), setting the speed of the lift, and scheduling stopping criteria for daily operation. Scheduling stopping criteria is an important task for elevators installed in businesses as well as in residential sites. Scheduling can be used to instruct an elevator to skip certain floors during peak hours. These scheduling instructions are sent in batch to the onboard controllers everyday.

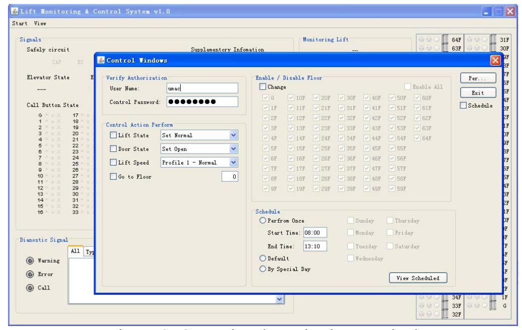
Figure 3. Control and monitoring terminal

The monitoring terminal pulls real-time status information of elevators from the central server for display. The monitoring terminal is crucial in observing real-time status of the individual elevator. Any error or emergency situations can be instantly displayed in the console.

VII. PLAYBACK

Playback is considered as one of the important functions in any elevator control and monitoring. Playback function (see Figure 4), allows maintenance engineers to reenact the sequence of activities during a specified time interval. Such capability is crucial in troubleshooting any hardware or software errors within the system.

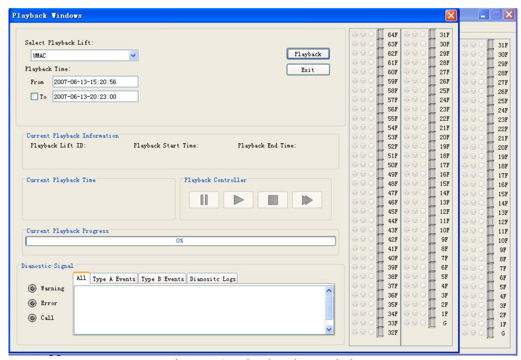
Figure 4. Playback module

Playback module extracts data (see Section IV) from the real-time database for displaying animation sequences. During a playback session (see Figure 4), the user is required to input the time interval for playback. Next, the playback module queries the real-time database to fetch the corresponding elevator status records which have been logged previously (see Figure 5). When the user presses the play button, elevator state records are retrieved from the real-time database server and converted it into strings for parsing. The output from parsing is then sent to the elevator states cache. Finally the Playback module fetches the elevator state snapshot from the cache and displays through the animation. The Playback module displays the animation of the elevator sequences according to a default speed.

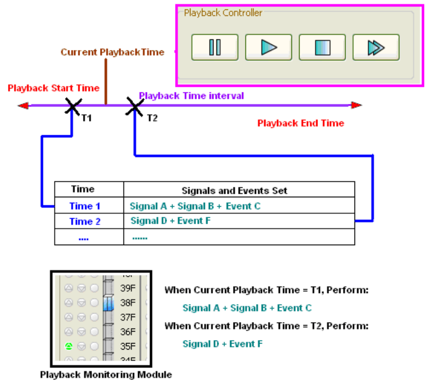

Figure 5. Playback design

VIII. MAINTENANCE SCHEDULING

Elevators are required to be regularly serviced by maintenance staff. In addition, urgent maintenance actions could be triggered by events which are generated from onboard elevator controllers. The genetic algorithm based scheduling module (see Figure 6) generates a daily/weekly maintenance timetable based on (a) the availability information of maintenance staff including on/off duty period, (b) events detected from elevators, and (c) predefined priority classification of events from the database.

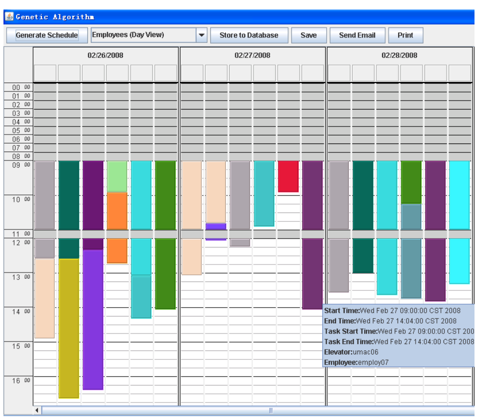

Figure 6. GA-based maintenance scheduling

In the genetic algorithm, an array with linked lists is used to represent a chromosome. During a typical cross-over operation, the resulted new chromosome may contain infeasible assignment of tasks and thus can violate constraints defined for a valid schedule. For instance, a chromosome may contain duplicate tasks or may assign tasks to a staff beyond his/her working hours. To alleviate these problems, we define restoration and mutation operations to correct faulty chromosomes within the population. A fitness function was also designed to assign higher degree of fitness to chromosomes which allow allocations of highest priority tasks at the beginning of the schedule.

IX. DATA WAREHOUSE DESIGN

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.

A Data Warehouse is "a subject oriented, integrated, nonvolatile, and time variant collection of data in support of management's decision" [7]. 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. In our system, a database is used 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. Against this background, we design a star schema which can be used to store relevant signals and maintenance records in a data warehouse[2]. 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. The star schema design is depicted in Figure 7 where a maintenance fact table is connected to three dimension tables: signal dimension, floor dimension, and lift information dimension. Based on the star schema, elevator signals are processed and stored based on interval rather than time instance. In Figure 7, 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 used to 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.

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

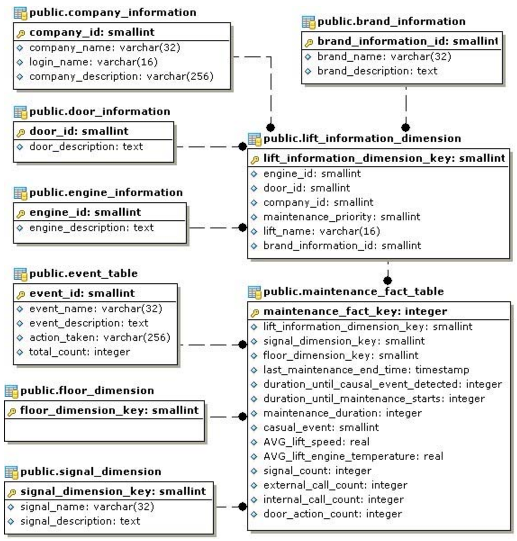

Figure 7. Star schema design for data warehouse

Based on the star schema in Figure 7, engineers are able to issue queries which can reveal information about the safety of the elevators and the effectiveness of the maintenance actions. Our prototype system shows significant reduction in storage requirement [12] and allows efficient query processing across massive real-time data.

X. CONCLUSION

In this paper, we describe the development of an event-driven elevator testing, control, and monitoring system. Our integrated system includes virtual controllers for testing and deployment, a playback function for troubleshooting, maintenance scheduling based on genetic algorithm for resource optimization, and a conceptualization of a star schema for transforming massive real-time data from traditional databases into multi-dimensional databases. The prototype system is currently being tested for future commercial release by our industry collaborator.

XI. ACKNOWLEDGMENT

This research is funded by the Research committee of the University of Macau and Hualong Fabrica De Elevadores Lda.

XII. REFERENCES

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

[2] OTIS Elevator Co., http://www.otis.com.

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

[4] H.M. Ryu, S.J. Kim, S.K. Sul, T.S. Kwon, K.S. Kim, Y.S. Shim, and K.R. Seok, 'Dynamic load simulator for high-speed elevator system,' in Proceedings of the Power Conversion Conference, page 885-889, vol.2, 2002.

[5] A. Monti, S.D. Arco, Y. Work, and A. Lentini, 'A Virtual Testing Facility for Elevator and Escalator Systems,' in Proceedings of Power Electronics Specialists Conference, IEEE, page 820-825, 2007.

[6] P. Cortés, J. Muñuzuri, and L. Onieva, 'Design and Analysis of a Tool for Planning and Simulating Dynamic Vertical Transport,' Simulation, 82(4), page 255-274, April 2006.

[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 and S.J. Ovaska, 'Delayless acceleration measurement method for elevator control', IEEE Transactions on Industrial Electronics, 45(2), page 364-366, April 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] B.A. Powell, 'The Role of Computer Simulation in the Development of a New Elevator Product', in Proceedings of the 1984 Winter Simulation Conference, page 445-450, 1984.

[12] Y.Y. Yang, Y.W. Si, and W.L. Leong. 'Data Warehousing Massive Real-time Elevator Signals and Maintenance Records', in Proceedings of 2008 IEEE International Conference on Industrial Technology (ICIT 2008), 2008, IEEE.

[13] Virtual Elevator Design, http://www.jobs-amst.com/virtual.aspx.

[14] A.J. Penz, 'A computer simulation approach to elevator system design.' In Proceedings of the 8th Workshop on Design Automation, ACM, page 259-268, 1971.

[15] M.L. Siikonen, Elevator Traffic Simulation, Simulation, 61(4), page 57-267, 1993.

[16] T.D. Barkand, 'Safe electrical design of mine elevator control systems,' IEEE Transactions on Industry Applications, 33(2), page 312-318, March/April 1997.

[17] T.D. Barkand and W.J. Helfrich, 'Application of dynamic braking to mine hoisting systems,' IEEE Transactions on Industry Applications, 24(5), page 884-896, September/October 1988.

[18] M. McCann and N. Zaleski, 'Deaths and Injuries Involving Elevators and Escalators,' The Center to Protect Workers' Rights, http://www.cpwr.com, Accessed 3 March 2008.

[19] S.C. Hui and F. Wang, Remote Video Monitoring Over the WWW, Multimedia Tools and Applications, 21(2), page 173-195, Springer, 2004.

[20] H. Yamashina and S. Otani, 'Cost-optimized maintenance of the elevator – single unit case,' Journal of Quality in Maintenance Engineering, 7(1), page 49-70, 2001.

[21] R.H. Crites and A.G Barto, 'Elevator Group Control Using Multiple Reinforcement Learning Agents,' Machine Learning, 33, page 235–262, Kluwer Academic Publishers, 1998.

[22] Y.C. Cho, Z. Gagov, and W.H. Kwon, 'Timed Petri Net Based Approach for Elevator Group Controls,' Transaction on Control, Automation, and Systems Engineering, 2(2), page 121-130, June, 2000.

[23] T. Beielstein, C.P. Ewald, and S. Markon, 'Optimal Elevator Group Control by Evolution Strategies,' Lecture Notes in Computer Science, LNCS 2724, page 1963–1974, Springer-Verlag, 2003.

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

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

[26] T. M. Nguyen, J. Schiefer, and 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): page 105-141, 2007.