

# Dyanmic Scheduling of Elevator Systems over Hybrid Petri Net/Rule Modeling

*Yan-Hau Huang and Li-Chen Fu*

Department of Computer Science and Information Engineering  
National Taiwan University, Taipei, Taiwan, R.O.C.

## Abstract

In this paper, a hybrid model of a multiple elevator system is proposed, consisting of a timed place Petri net (TPPN) model and a set of control rules implemented via the so-called control places in the TPPN model. The Petri net model is a highly modulized structure, whose constituent modules can be classified into four types: Basic Movement Module, Loading/Unloading Module, Direction Reversing Module, and Call Management Module. The whole complete model is a combination of the copies of the above four modules. Since the firing sequences of the TPPN equate the evolution of the modeled system, they can be regarded as a schedule. A dynamic scheduling strategy is proposed to obtain the desirable schedule. A new concept of control places is also introduced in the proposed model so as to make the modeling more precise and to reduce the reachability graph more efficiently. To show the feasibility of the proposed method, an emulator of the elevator system is constructed for demonstration.

## 1 Introduction

An elevator system is used to supervise multiple elevators and to ensure that they are operated efficiently. The elevator system must decide which of several elevator cars should respond to a call made by passengers, in order to provide the highest possible level of services. To build a good elevator system for solving the elevator scheduling problem, we need to build an elevator model first to represent the cooperation of multiple elevators. Petri net has evolved into an elegant and powerful graphical modeling tool for asynchronous, concurrent event-driven system and, therefore, is chosen here as a suitable modeling tool.

However, the elevator system is usually involved with human behaviors and preferences. Very often,

there may not exist an optimal scheduling solution and, hence, the control system must be flexible enough to handle all possibly incoming situation. In view of this, a hybrid Petri net model is thus introduced in this paper to serve the purpose, i.e., besides the conventional timed place Petri net model, several heuristic rules and control places are introduced to enrich the flexibility of the aggregate model. Specifically speaking, these additional rules in the model will affect the general behavior of the Petri net through direct adding/removing of tokens in the control places. Upon the occurrence of an external event, the model will search for the feasible rules to adjust the marking of the resulting Petri net, whereby the hybrid model will run properly to produce an adequate scheduling result to handle that situation for service requirement.

Petri net [1, 2, 3, 4, 5] is a graphical and mathematical modeling tool applicable to many systems. They are promising for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. Petri net is useful tool for modeling [6, 7], simulation, and scheduling discrete event systems such as flexible manufacturing systems.

There are researchers who proposed some colored Petri net to model real-time system and used the elevator system as examples [8, 9]. But these models are only for one elevator and are in lack of the coordinator part for a whole elevator group control system. Due to the fast development of computer technology, integrated circuit(IC) and micro-computer are widely used in modern elevator system. The computer technology introduces the artificial intelligence(AI) technique into the elevator group supervisory control system [10, 11]. Moreover, the fuzzy control logics [14] and expert systems are also used to analyze the traffic pattern and to predict the possibly incoming call requests [11, 12, 13]. Conceivably, the utilization of the AI and expert sys-

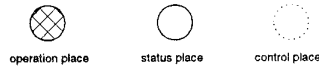


Figure 1: Types of places

tems in the context of elevator system control is an indispensable trend.

## 2 Modeling and Scheduling

### 2.1 Systematic Modeling

#### Problem Description

In the elevator system, there are two types of calls. The *hall call* is given through buttons on the hall of the building, and the *car call* is given in the elevator by the passengers. An elevator system has a pair of hall call buttons on each floor, one for *up hall call* and the other for *down hall call*. As soon as a hall call button is pressed, the elevator system must register the hall call, and selects and assigns an elevator to serve the hall call. After serving the hall call, the passenger should press car call button to register his destination floor and the elevator must move up/down to stop at destination floor. When all calls are served, the elevator stops and waits for the next call to arrive, and then repeat the above operation again.

In this thesis, we select the TPPN to model our elevator system as has been mentioned in the previous chapter. In this TPPN model, all the places can be classified into three categories listed below, as shown in Fig 1:

- 1) **Operation places:** These places are timed places. A token at an operation place represents a specific operation is now being performed.
- 2) **Control places:** Control places are used as an interface between the environment and the kernel, which represent the signals received from the world outside the elevator system or from the elevator system.
- 3) **Status places:** These places are used to record some specific information about the current elevator system.

Due to the complexity of describing various kinds of behaviors of a groups of elevators, it is hard to construct the Petri net model directly. Therefore, we first

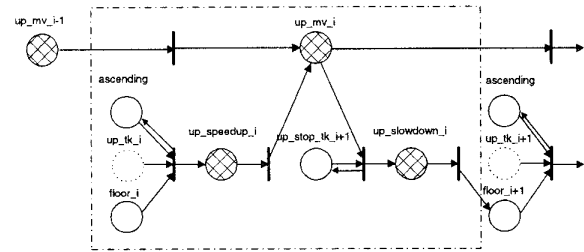


Figure 2: Basic Ascending Movement Module

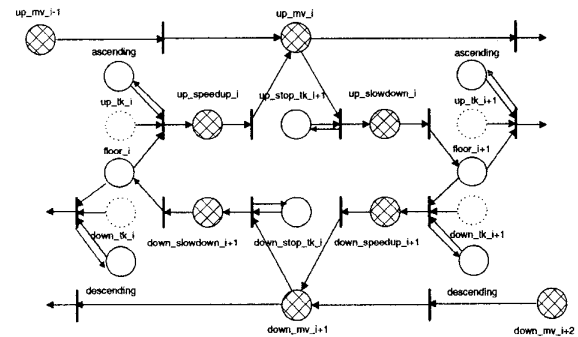


Figure 3: Basic Movement Module

observe the architecture of the elevator hardware and analyze the activities of each elevator, and then build four kinds of macro modules so that the final Petri net model can be constructed from these macro modules.

#### Basic Movement Module

The basic movement module is the most fundamental part of the Petri net model of a group of elevators. It describes the movement activities of each elevator between floors. The Petri net model for up-movement is shown in Fig 2. Fig 3 shows the combined module of up-movement and down-movement. By repeating this strategy for every suffix  $i$ , we can eventually construct the full movement model of an elevator car. Then, by adding a prefix serial number, we can represent different elevators. All places without suffix represent that they are shared by this elevator. Control places in this module and the following module will be explained in the section of Control Rules detailedly.

#### Loading/Unloading Module

The Loading/Unloading Module describes the event that passengers get into or get out of the elevator car. An elevator must be parked at the floors designated by the car calls, but it may not stop at the floors

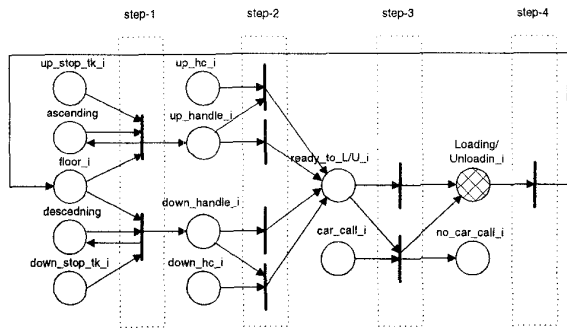


Figure 4: Loading/Unloading Module with firing steps

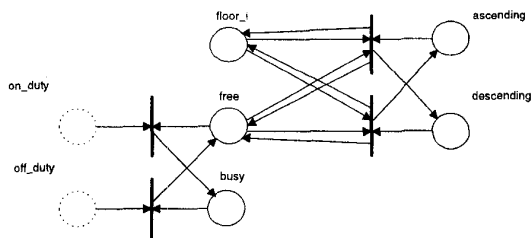


Figure 5: Direction Reversing Module

designated by the hall calls for some reasons. The Loading/Unloading Module is shown in Fig 4. Places in different modules but with the same notations represent that they are shared by those modules. One can interconnect the different modules via the shared places.

### Direction Reversing Module

Direction Reversing Module describes the timing to reverse the moving direction of an elevator car, which is shown in Fig 5. We explain the notations as follows:

- 1) **on\_duty**, represents that the elevator is on duty. Once the scheduling system decides to dispatch an elevator car to respond to an up-hall call, it is not reasonable to dispatch this elevator to serve another down-hall call before sending the passenger(s) who has(have) made the up-hall call to his destination. This control place is thus needed to inform the scheduling system to prevent such a case.
- 2) **off\_duty**, is mainly used to switch the status of an elevator from busy to free. "off\_duty" means that there is no higher car call requesting for service when the elevator is ascending, and there is no lower car call requesting

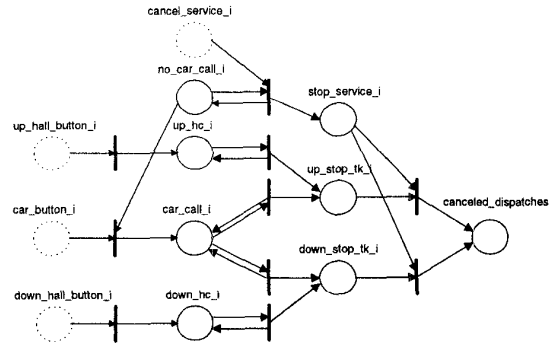


Figure 6: Call Management Module

for service when the elevator is descending. When an elevator is off duty, this is a suitable timing for the elevator car to reverse its moving direction.

- 3) **free**, represents that this elevator has no car call now. It can reverse its direction if necessary.
- 4) **busy**, represents that this elevator is busy serving some calls. If a token is placed in this place, this elevator is denied to reverse its direction.

To check whether an elevator car can reverse its moving direction, we must check all hall calls and car calls in its moving direction. In order to simplify the model, we use control places "on\_duty" and "off\_duty" to replace the checking model.

### Call Management Module

Call Management Module is mainly used to manage the car calls and the hall calls. When the system finds that service of some hall call may be a better choice, it is a good opportunity to cancel the service and re-assign the hall call to another elevator car. We show the Call Management Module in Fig 6. To make distinction with other modules shown above, there will be no prefix serial number with "up\_hall.button\_i", "down\_hall.button\_i", "up\_hc\_i" and "down\_hc\_i" places. These four kinds of places are common places for all elevators. Through these places, different elevator models will be linked together. An example of this will be shown in the next section.

### A Simple Example : Modeling with Modules

In Fig 7, we show a simple example. There are three elevators in a building with four floors in the



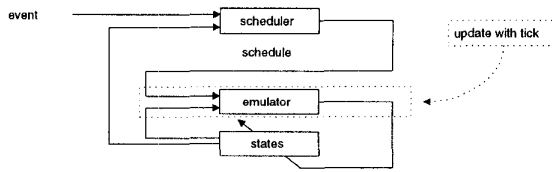


Figure 8: Block diagram of the dynamic scheduling.

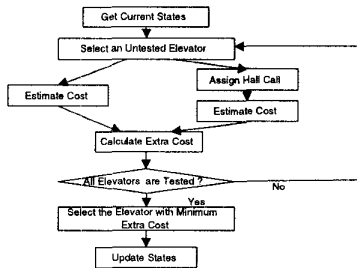


Figure 9: Flow of the hall call assignment.

triggered every fixed time interval, say, a second, by the system clock. Here, we also use *tick* as the time unit for tracing the transition of firings of our TPPN model. The block diagram of the system is shown in Fig 8.

To provide satisfactory performance, we embed the sense of satisfaction of passengers into our elevator system, estimate detailedly the cost by simulation over our Petri net model, and refine the dispatching by an refinement algorithm.

### Cost Functions

For passengers, when the waiting time exceeds the endurable limit, the cost will grow rapidly. And, when the riding time exceeds the reasonable limit, the passengers may lost their patience. We try to simulate the real circumstance of elevator system by estimating cost with some accumulative cost functions. We take waiting time, riding time, and power consumption as parameters of our cost estimation. In order to reduce the probabilities of unendurable long waiting or unreasonable long riding, the cost of waiting is piecewise linearly increasing with time, and the cost of riding is considered with both the riding time and the moving distance. We define a aggregate weight cost :

$$cost = waiting\ cost + riding\ cost + power\ cost.$$

### Hall Call Assignment

In our system, when a hall call occurs, we first assign the hall call to an elevator randomly. Then, we

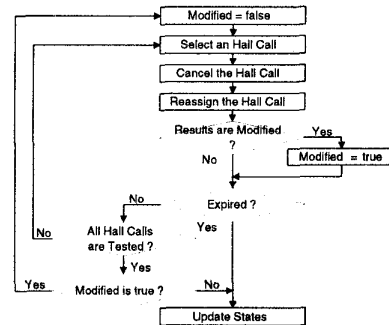


Figure 10: Flow of the refinement algorithm.

simulate the consequence of the assignment by evolving from the current marking to a goal marking over our Petri net model, and record the total evolution cost. After that, we assign the hall call to another elevator and repeat the above steps. After the exhausted simulations, we choose the elevator with the minimum cost as our preferable dispatching. Fig 9 shows the flow of the hall call assignment.

### Refinement

When any of the aforementioned events occurs, the prediction made in hall call assignment should be updated, and the dispatch should be updated as well. To resolve the problem, we first make an implicit assumption that, whenever an event occurs, the influence to the optimality of the previous schedule should be local, i.e., the previous schedule should not be far from the optimal schedule. Thus, we try to correct our schedule to approach the optimal schedule. However, such assumption may not adequate, but it is hardly possible to schedule the whole system again in a real-time system. Hence, we propose an algorithm that focuses on the solution set near our previous schedule to search for a refined one. Fig 10 shows the flow of the refinement algorithm. Since the hall call assignment offers us a local optimal dispatching, it is a good idea to refine the dispatching rather than to perform a global search. The algorithm is efficient and economic, because it only takes constant memory and occupies little CPU times. Such desirable performance can be observed in the simulation results.

### Flow of Dynamic Scheduling

Fig 11 shows the flow of dynamic scheduling. To approach the goal of real-time dynamic scheduling, we propose an event-driven dynamic scheduling strategy.

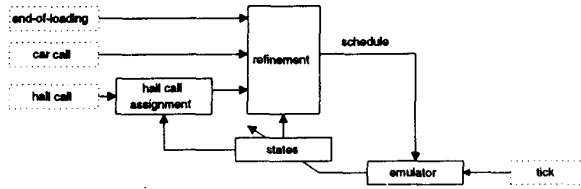


Figure 11: Flow of dynamic scheduling.

To reflect the nature of human's satisfaction about an elevator service, we choose accumulative cost functions. The cost functions can help to reduce the probability of long waiting time efficiently. The hall call assignment method computes the precise cost by simulation over our TPPN model, and the refinement algorithm is used to reschedule the elevator dispatches easily and efficiently.

### 3 Conclusion

In this thesis, we proposed a systematic Petri net based modeling for an elevator system. The objective of Basic Movement Module is to model the trajectories of an elevator moving from one floor to another. The Loading/Unloading Module is used to deal with call states and activities of loading/unloading. The Direction Reversing Module is designed to handle the suitable timing to reverse the moving direction of an elevator. And the Call Management Module is used to manage the hall calls and car calls, and to integrate different elevators into a group. In these modules, the control places are utilized to simplify the complexity of the Petri net model, and the control rules are used to satisfy human nature and to approach the goal of efficiency.

To obtain an efficient schedule, we propose an event driven dynamic scheduling strategy, and corresponding algorithms. The strategy helps us to handle an elevator system with insufficient information and uncertain events. We estimate the cost of dispatching by Hall Call Assignment Method. The assignment method precisely computes the cost over our TPPN model. And the refinement algorithm is simple and efficient to refine our schedule in a real time system.

### References

[1] C. A. Petri, "Kommunikation mit Automaten," Ph.D. Dissertation, University of Bonn, Bonn, West-Germany, 1962.

[2] James L. Peterson, "Petri nets," *Comput. Surveys*, vol.9, Sep. 1977.

[3] James L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, INC., Englewood Cliffs, N.J. 07632, 1981.

[4] T. Murata, "Petri Nets : Properties, Analysis and Applications," *Proc. IEEE*, vol.77, NO.4, pp.541-580, Apr. 1989.

[5] Richard Zurawski and MengChu Zhou, "Petri Nets and Industrial Applications: A Tutorial," *IEEE Trans. on Industrial Electron.*, vol.41, pp.567-583, Dec. 1994.

[6] Chin-Jung Tasi, Li-Chen Fu and Yung-Jen Hsu, "Modeling and Simulation for Flexible Manufacturing Systems Using Petri Net," *2nd Int. Conf. on Automat.*, vol.4, pp.31-38, 1992.

[7] Chu-Hui Lin and Li-Chen Fu, "Petri Net Based Dynamic Scheduling of an Elevator System," *IEEE Int. Conf. on Robotics and Automat.*, vol.1, pp.192-199, 1996.

[8] R. K. Guha, S. D. Lang, M. Bassiouni, "Software Specification and Design Using Petri Nets," *Proc. 4th Int. Workshop on Software Specification and Design*, pp. 225-230, 1987.

[9] F. S. Etessami, G. S. Hura, "Rule-Based Design Methodology for Solving Control Problems," *IEEE Trans. on Software Engineering*, vol.17, no.3, pp. 274-282, 1991.

[10] N. Kameli, K. Thangavelu, "Intelligent Elevator Dispatching System," *AI Expert*, pp. 32-37, Sep. 1989.

[11] H. Aoki, K. Sasaki, "Group Supervisory Control System Assisted by Artificial Intelligence," *Elevator World*, pp. 70-80, Feb. 1990.

[12] S. Tsuji, M. Amano, S. Hikita, "Application of the Expert System to Elevator Group-Supervisory Control," *IEEE 5th Conf. of Artificial Intelligence Applications*, pp. 288-294, 1989.

[13] T. Tobita, H. Inaba, K. Yoneda and T. Ueshima, "An Elevator Characterized Group Supervisory Control System," *IEEE IECON'91*, pp. 1972-1976, 1991.

[14] Chang Bum Kim, Kyoung A. Seong, Hyung Lee-Kwang, Jeong O. Kim, and Yong Bae Lim, "A Fuzzy Approach to Elevator Group Control System," *IEEE Trans. Syst. Man Cybern.*, vol. 25, NO. 6, pp. 985-990, Jun. 1995.