Proceedings of the 5th World Congress on Intelligent Control
and Automation, June 15-19, 2004, Hangzhou, P.R. China

第五届全球智能控制与自动化大会
2004 年 6 月 15 - 19 日，中国杭州

A New Modeling Method for Elevator Group Control System with Cellular Automata*

Yuge Xu, Fei Luo and Jianguo Wang

College of automation
South China University of Technology
Guangzhou, Guangdong Province, China

xuyuge@163.com

***Abstract* - Aiming at elevator group control systems (EGCS), this paper proposes a new modeling method with cellular automata theory and builds a EGCS model which consists of three modules: group control module, elevator movement module and assignment sequence module. Different group control principles can be applied in the group control module. Elevator movement module simulates the moving behaviors of elevators by using cellular automata. Assignment sequence module is established to logically store call orders to every elevator. This model is simply structured and easy to modify. Computer simulations show that the proposed modeling method is creditable and the established EGCS model gives out satisfied performance. This modeling method can also be applied to other discrete event dynamic systems.**

Index Terms* - *Elevator group control system, cellular automata, assignment sequence, simulation.

I. INTRODUCTION

The elevator group control system (EGCS) is a discrete event dynamic system that systematically manages three or more than three elevators in order to transport the passengers efficiently. It is complicated, nonlinear and stochastic in time and space. With the development of high-rise buildings in recent years, the level of performance and quality of service that EGCS should possess is increasing. In order to find effective elevator group control methods and verify these methods, we first need to set up a model of EGCS to describe elevator dynamic behaviours and the cooperation of elevator groups. A good EGCS model is very important to study multiple elevators control technique. It should have correct simulation results and simple structure that is facile to modify.

In recent years, many theories have been applied to establish EGCS model as in reference [1]~[5]. Time Logic theory and Petri net are in common use. These modeling methods have gained many achievements. But with car number and building floor number increase in the model, the structure of the whole model inevitably becomes complicated. Along with it, the model simulation program will become larger and larger. It will enlarge system errors and even make the whole EGCS model unreliable.

Cellular automata theory was first proposed by John von Neumann in 1940s. Reference [6]~[10] introduced this theory in detail. Cellular Automata is discrete, stochastic and parallel-distributed. These characteristics allow it to describe complex system perfectly. It is a useful tool to model and simulate the discrete event dynamic system. The outstanding characteristics of cellular automata in modeling and simulation allow us to apply cellular automata theory to the model of EGCS. In the anticipant cellular automata model of EGCS, each cell is correlative only with its neighbouring cells but independent with the others. The increase of cells number in the model does not affect the structure of the whole model, and therefore it will not enlarge system errors. Hence a model with cellular automata can be easily modified or extended, we can discretionarily add or reduce building floor number and car number by modifying a small portion of codes in the model program. Moreover, in the cellular automata model, the number of the cell rules is finite, it not only makes the whole model simple but also makes simulation program short and easy to realize.

In Section II, we provide an introduction of cellular automata theory. In Section III, a description of our proposed EGCS model is given in details. In Section IV we show the simulation results and finally a conclusion is made in Section V.

II. CELLULAR AUTOMATA

Cellular automata consists of four elementary parts: a regular discrete lattice of cells, state space of cells which has a finite set of states, a finite number of neighbouring cells and the same local rules to every cell. The evolution takes place in discrete time steps and the neighbourhood relation of each cell is local and uniform. According to the same local rules, every cell evolves synchronously depending on the states of the own cell, the states of the finite number neighbouring cells and external input. Cellular automata also can be defined using mathematics language:

Let L_d be a regular lattice (the elements of L we call cells), d be the dimension of cells space, S be a finite set of states, N be a finite set (of size $|N|=n$) of neighbourhood indices such that for all c in N, all r in L_d : $r+c$ in L, $N=(s_1,s_2,...s_n)$, $s_i \in Z$, $i=1,2,...n$, f: $S^n \rightarrow S$ a transition function.

Then we call the 4-tuple (L_d,S,N,f) a cellular automaton.

* Foundation items: Supported by the National Natural Science Foundation of China (69684001).

0-7803-8273-0/04/$20.00 ©2004 IEEE

III. MODELING of EGCS

In this part, we present an EGCS model and interpret the application of cellular automata in the EGCS modeling process.

Fig.1 is an overview of our EGCS structure. The proposed system is composed of three modules: group control module, elevator movement module and assignment sequence module. The task of the group control module is selecting a suitable elevator for each hall call in terms of the system information (including traffic flow, whether over loading or not and the current moving situation of each car). There is a model of multiple elevators in the elevator movement module, which describes elevator's dynamic moving behaviours by cellular automata. Assignment sequence module logically stores hall call assignment orders and car call orders. Elevator movement module is connected to group control module through assignment sequence.

A. Elevator Movement Module

From the system structure, we know that every elevator has the same moving mechanism and is independent to each other. So, we set up the same model for each elevator. Multiple elevators model could be realized by operating several elevator models synchronously. We have analyzed elevator's moving behaviours and found that the movement of every elevator car could be divided into several parts: running (including acceleration, uniform speed and deceleration), waiting (including open/close door, load/unload passengers) and stopping (means no call order).

The cellular automata model takes place in a fixed discrete time step. We regard *T*, the minimum periodical time of the cellular automata evolution, as our system period, which depends on the particular elevator movement parameters and the model's precision requirement.

At first, we define some parameters needed in our model: (See Fig.2)

Layer, elevator car current floor, *t* current moment, *t+T*, next moment;

S[layer][t], the current state of the cell;

S[layer-1][t], the state of the lower level neighbouring cell;

S[layer+1][t], the state of the upper level neighbouring cell;

Assignment[], assignment sequence, see part B.

Cellular automata model definition is as follows:

1) Dimension: the dimension of cells space is 1 and building floor number is *n*;

2) States: each cell has 4 states. *s[layer][t]=0*, cell empty; *s[layer][t]=1*, cell with car and no movement; *s[layer][t]=2*, cell with car and moving up; *s[layer][t]=3*, cell with car and moving down;

3) Neighbours: each cell has two neighbouring cells. In Fig.2 (b), cell *i* has one neighbouring cell *i+1* above it and the other *i-1* under it;

4) Local rules:

As we know in practice, one elevator has only one car. It means that in the single cellular automata model there is only one cell denoted to be a car and the other cells are empty. The state of the cell with car at the next moment is only correlated with the state of this cell, the states of its two neighbouring cells and the external input but no correlation with other cells. So, other cells keep their former states during the cellular automata model evolution. Assignment sequence is model's external input, and it controls every moving behaviour of the elevator car. In each evolutional step, assignment sequence puts one call order into elevator movement module, which indicates the current call's destination floor. Movement module compares the destination floor with the current floor of the car, and then decides the next state of the cell based on the local rules.

Local rules for cells in detail are shown as follows:

a) When *s[layer][t]=1*, the cell with car and not moving at time *t*: If at the next moment *t+T*, waiting time is over and no new call order in the assignment sequence, the car stops, *s[layer][t+T]=1*; if waiting time is not over, the car continues to wait, *s[layer][t+T]=1*; if waiting time is over and assignment sequence has new call, the car moves up or down according to the call's direction, *s[layer][t+T]=0, s[layer+1][t+T]=2* or *s[layer][t+T]=0, s[layer-1][t+T]=3*.

b) When *s[layer][t]=2*, the cell with car and moving up at time *t*: if at the next moment *t+T*, the car doesn't reach the destination floor, it continues going up, *s[layer][t+T]=0, s[layer+1][t+T]=2*; if the car reaches the floor, it stops there and begin waiting, *s[layer][t+T]=0, s[layer+1][t+T]=1*;

c) When *s[layer][t]*=3, the cell with car and moving down at time *t*: if at the next moment *t+T*, the car hasn't reached its destination floor, it continues going down, *s[layer][t+T]=0, s[layer-1][t+T]=3*; if the car reaches the floor, it stops there and begins waiting, *s[layer][t+T]=0, s[layer-1][t+T]=1*;

d) When *s[layer][t]=0, s[layer-1][t]=0* and *s[layer+1][t]=0*, the cell and its neighbouring cells are all empty at time *t*: At the next moment *t+T*, these three cells keep their current states.

There are some special rules for the cells in the boundary. The cell with car on the top floor can only move down and the cell with car on the bottom floor can only move up as in practice.

Fig.2 (a), (b) and (c) shows a car in the elevator running down in uniform speed from floor *i+1* to floor *i-1* during time *t* to *t+2T*.

In the elevator movement module, there also defines a parameter *over_loading*, which is a signal to report whether the current car is over loading during the moving process.

B. Assignment Sequence Module and Flow Chart

In the system model, there are two types of call. Hall call is given through buttons on the hall of the building, and car call is given by the passengers inside the elevator car. Assignment sequence is built for each elevator. It logically stores the hall call orders and car call orders, which are assigned to the elevators. When the car begins to move, it takes out the first order from its assignment sequence and as soon as the car reaches the appointed floor, this order is cancelled from the assignment sequence. And when a new hall call has been assigned to an elevator, the corresponding assignment sequence will update immediately.

The flow chart of EGCS model is shown in Fig.3. It interprets the operation process of the whole model. When a hall call occurs, it goes into group control module. Group control module begins to calculate according to certain group control arithmetic and then selects the optimal elevator to respond this hall call. If the hall call is served, it will trigger one or several car calls. And while a car call occurs, it is logically put into the corresponding assignment sequence straightly. Stop station information is recorded if the car finished serving a call.

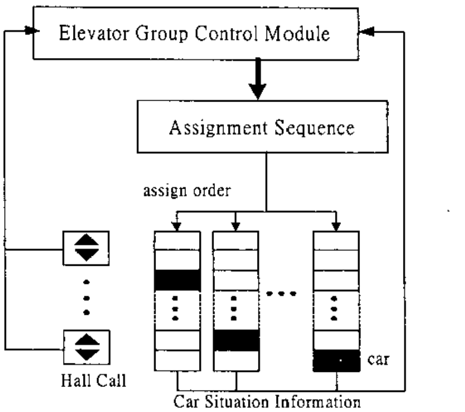

Fig.1 System structure

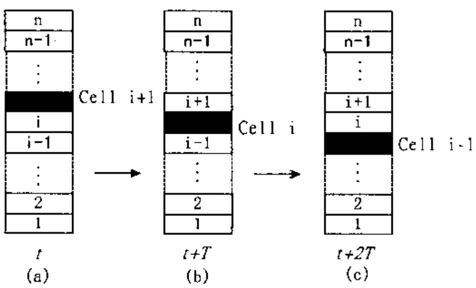

Fig.2 Cellular automata model structure

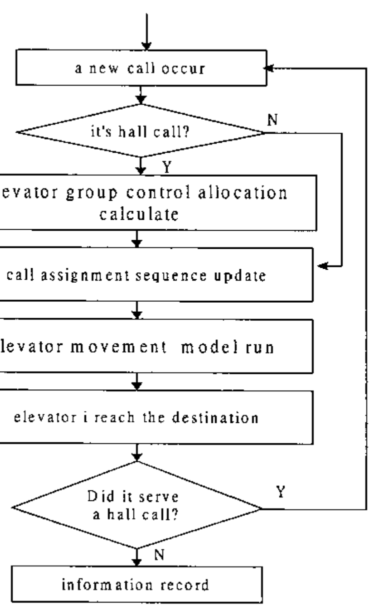

Fig.3 EGCS model flow chart

IV. SIMULATION

We implemented a computer simulation to verify the performance of the proposed model. Table I lists the simulation conditions.

The data of passenger traffic flow is simulated following the routine method in reference [11]. The number of passenger arrivals in random interval is generated through a Poisson process. Monte Carlo examination generates the data that from which floor the hall call issued and to which floor passenger wants. Here, we chose the minimum average waiting-time algorithm as our group control method.

To interpret the operation status of the EGCS model, at first we built a simple model, which only have 3 elevators and 10 floors. In Fig.4, 3 cars respectively stop at 1,4,7 floor at first and 10 hall calls are input the simple model in 80 seconds.

Then we operated the EGCS model in three traffic conditions. Table II and III show the simulation results. Average waiting time, average riding time and average serving time are defined as AWT, ART and AST.

Inter floor mode 50 persons/5min
Up-peak mode 150persons/5min
Down-peak mode 150persons/5min

Table I SIMULATION CONDITIONS

Items	Settings
Moving speed	2m/s
Acceleration/deceleration	1 m/s^2
Open/close door time	3s
Average load/unload time	3s
Average height of floor	3.3m
Capacity	15 persons

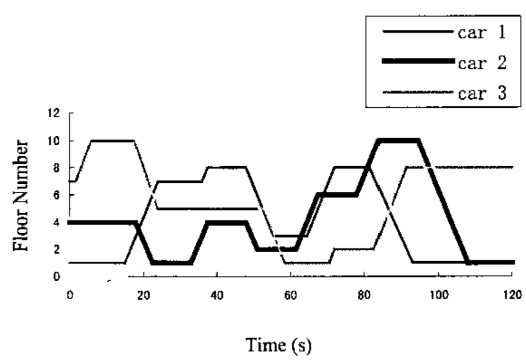

Fig.4 Simulations of a simple model with 3 elevators and 10 floors

Table II
15 Floors 4Elevators

Traffic mode	AWT	ART	AST
Inter floor	14.9s	33.6s	48.5s
Up-peak	29.1s	49.3s	78.4s
Down-peak	33.5s	52.1s	85.6s

Table III
20 Floors 6Eelevators

Traffic mode	AWT	ART	AST
Inter floor	15.2s	34.5s	49.7s
Up-peak	31.4s	51.6s	83.0s
Down-peak	32.8s	53.1s	85.9s

These simulation results demonstrate that the proposed EGCS model based on cellular automata can not only accurately describe the dynamic behaviours of the multiple elevators but also operate correctly in the different traffic conditions. It is proved that this modeling method for EGCS is effective.

V. CONCLUSION

In this paper we have proposed a new modeling method for EGCS. The model of EGCS that takes cellular automata theory into account has been built. It consists of three modules: group control module, elevator movement module and assignment sequence module. Computer simulations confirmed the validity of the proposed method. The EGCS model could operate correctly in different traffic flow conditions.

The study of applying cellular automata theory to EGCS model is just at the beginning. But, the outstanding characteristics of cellular automata make it to be a strongly useful tool in the modeling fields of EGCS. This modeling method also can be applied to other similar discrete dynamic systems.

REFERENCES

[1] Dyck Derek N and Caines Peter E, "The logical control of an elevator," *IEEE Trans. on Automatic Control*, Vol. 40, no.3, 1995.

[2] C. Lin and L. Fu, "Petri net based dynamic scheduling of an elevator system," *IEEE Int. Conf. On Robotics and Automat*, Vol.1, pp.192-199, 1996.

[3] Y. Ho and L. Fu, "Dynamic scheduling approach to group control of elevator systems with learning ability," *IEEE Int. Conf. On Robotics and Automat*, Vol.1, pp.2410-2415, 2000.

[4] Q. Zong, X. Shang, Y. Yue and X. Lei, "Virtual environment design of elevator group control system," *Manufacturing Automation*, Vol.21, no.5, pp.24-25, 1999.

[5] Q. Zong, Y. Cai and X. Lei, "Object oriented petri net based modeling of elevator group control system," *Systems Engineering and Electronics*, Vol.23, no.1, pp.27-30, 2001.

[6] J. Li, "An approach to mechanical analysis of structures of bar system using cellular automata." Wuhan University of Technology, Master's Thesis, March 2002.

[7] W. Jin, J. Zhang and Y. Zheng, "Traffic flow simulation model based on cellular automata," *Journal of South University of Technology*, Vol.29, no.8, pp.93-96, August 2001.

[8] M. Wang and X. Geng, "JAVA simulation of cellular automata," *Journal of Dalian Railway Institute*, Vol.23, no.2, pp.56-58, June 2002.

[9] X. Jin, "The simulation of cellular automata based on matlab," *Computer Simulation*, Vol.19, no.4, pp.27-30, July 2002.

[10] Z. Yu, S. Jiang, J. Diao and W. Xie, "Modeling and simulation of elevator group control system based on cellular automata," *Shanghai Electric Technology*, Vol.1, 2002.

[11] S. Guo and L. Li, "Elevator passenger traffic and video screen simulation," *Journal of Taiyuan Institute of Machinery*, Vol.12, no.1, pp.38-43, 1991.