A Simulation Study on Public Building's Staircase Fault Tolerance

Hong-Sheng CHEN and Feng-Tyan LIN
Graduate Institute of Building and Planning, National Taiwan University, Taiwan

Keywords: agents, fault tolerance, simulation

Abstract: This paper applies the concept of fault tolerance to staircase layout plan. Fire or smoke may cause some staircases in a building inaccessible. We argue that architects should assess the result from the possible fault of vertical routes. The capability of tolerating staircase faults depends on space usage, arrangement, and pedestrian attributes. In this study, a mathematical model is constructed. For analysing pedestrian's movement in interior space, we employ Monte Carol simulation and Agent-Based Modelling method in a CAAD environment. It helps us to visualise the dynamic process of agent's evacuation process, and to test the problem of possible staircase faults. Finally, a case study brings some important discoveries.

1 INTRODUCTION

Fault tolerance is an important issue for a system designer, who has responsibility to make sure that system has enough ability to resolve component's fault (Anderson and Lee 1981). This issue has been studied in computer science, mechanics, and commercial research for a long time (Karyagina 1997), but there are little researchers ever explore its meaning and importance in architecture. In this article, we will apply the concept of fault tolerance to examining the design of staircases in public buildings when they are suffering from fires, earthquakes, accidents, or any emergency situations.

The layout problems have been studied for a long time (Jo and Gero 1998; Liggett 2000). In particular, staircases are essential components in building layouts. Architects usually check the appropriateness of staircases according to building codes in terms of size, amount and location (Templer 1992). However, building codes regulate general cases. Designers still need some software tools to evaluate staircase layout for them. Furthermore, simulation methods that can show the performance of staircases in a dynamic way are particularly expected. For example, if any one of the staircases is failed in an emergency situation, does it take too long to evacuate people through other staircases? Will "backup staircase" work as designers expect? How to layout the staircases such that they serve roughly equal amount of people? These questions are especially important for public buildings with a huge crowd of people.

In the following sessions, we will introduce an agent based model (Jonker and Treur 2001) to simulate the evacuation behaviour by randomly setting fires and failing staircases. Besides, some measurements are proposed to evaluate the performance of staircases in terms of fault tolerance.

2 RESEARCH METHOD

We use an agent-based model to simulate the performance of staircase layout. Agents (people) in the model have various properties, such as movement speed and response time. Agents independently but interactively move in the system. Accordingly, designers evaluate the performance of the system (staircase layout in this research) by measuring some criteria.

2.1 Declaration

1) S: A situation, which describes a floor status of spaces and pedestrian.
2) N: A number, which notes the number of staircases in S.
3) T : A period of time, which is the total evacuation time for all crows in S without any failed staircase.
4) 4. T^k : There are k staircases fail to work, and T^k is the total evacuation time. (0<k<N)
5) α : Let $\alpha = \frac{T^k}{T}$, α shows the increase time proportion.
6) D: D = {d | d is all the staircases on a floor}, i, j, … $\in$ D.
7) $T_{i,j..}$: i , j,…are the failed staircases, T^k could be explicitly rewritten as $T^k_{i,j...}$.
8) $\alpha_{i,j..}$: Rewriting fifth declaration as $\alpha_{i,j..} = \frac{T_{i,j..}}{T}$.
9) ΔT : It shows evacuation time increased.
10) $$\Delta T = T^k - T = T_{i,j..} - T = (\alpha - 1)T$$
10) Max ($T^k_{ij...}$, $T^k_{jh...}$, …) : It is the maximum of $T^k_{ij...}$, $T^k_{jh...}$,…

2.2 Tolerance of Evacuation Time

1) FT^k -- the maximum evacuation time requested by building code. Design constraint: Max $(T^k_{ij\ldots}, T^k_{jh\ldots}, \ldots) < FT^k$

2) α^k -- the expected maximum time ratio of faulty situation to normal situation.

   Design constraint: $\forall$ i, j, … $\in$ D, $\frac{T_{i,j..}}{T} < \alpha^k$

3) NT -- a relative requirement about two conditions (k=0 and k>0).

   $N \cdot T^n \geq (N - k)T^k$, or $T^k \leq \frac{N}{(N-k)}T^n$

2.3 Tolerance of Crowd Burden

When some staircases fail to work, it may or may not increase another staircase's burden. If X fails, it increases Y's burden, the pedestrian increase from Y_0 to Yx, then we express this phenomenon as:

$\beta_y = \frac{Yx}{Y_0}$, β_y is the pedestrian change ratio of Y.

Having every staircase serves roughly equal amount of the pedestrians is a reasonable consideration in design. However, staircase faults may destroy such balance. We calculate the standard deviation of the amounts of pedestrians associated with staircases to measure the degree of balance. A small standard deviation means that pedestrians use staircases more evenly.

Let x1, x2…xn be the amount of pedestrians using staircase i, their SD (standard deviation) is:

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}} \qquad where\ \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

Let Sn be SD in normal situation, and Sn' be SD in faulty situation.

$$\Delta S = |Sn' - Sn| \tag{2}$$

Equation 2 measures the degree of imbalance among staircase burdens. A high

D*S* implies the usage of staircases is imbalanced.

3 FAULT MODEL

It is essential to construct a proper fault model (Anderson & Lee, 1981). This section introduces a theoretical model for pedestrian movement on a floor space.

3.1 Modelling Space, Object and Activity

Declaration:

1. Environment Set	E : environment
2. Objects Set	X = {x_1, x_2…, x_n} : the set of all the pedestrians
	O É {wall, column, obstacles} = {o_1, o_2…, o_n}
3. Space Set	D = {d_1, d_2…, d_n} : the set of all the staircases
	Sa É {room, toilet, space with door} = {Sa_1, Sa_2, …, Sa_n}
	Sb É {corridor, passage without door} = {Sb_1, Sb_2, …, Sb_n}

E = (Object Sets) È (Space Sets) = (X È O) È (D È Sa È Sb)

3.1.1 Symbols and Predicates

x: a pedestrian in E, x Î X.

Vx: velocity of x's movement.

Tx: time of x starts its movement.

Wpq: an obstacle (wall, column) to x, Wpq Î O.

p and q: two ends of Wpq.

Dx: the nearest staircase to x, Dx Î D.

R(x, a, b): x stands at room a, and the nearest door to x is b, a Î Sa.

Mx: distance of x's movement.

Nx: direction of x's movement.

Px: x's position.

Px': x's position in next period of time.

t: a period of time.

T: a time process of simulation.

D(i, j): direction from i to j.

L(i, j): distance from i to j.

Occupy (x): $Px' = Py, y \neq x$.

Between (Px, Px', o): between Px and Px', $\exists\, o \in O$.

$P(x, y) = a\,\%, x \in X$: pick a x from X, the probability of x has attribute y is a%.

3.1.2 States

For every pedestrian, there are four states describing his/her starting, position, waiting, and obstacles. These states are used for showing x's movement states and for setting movement rules. State_a compares Tx and T. State_b shows x's position. State_c checks if there is any other pedestrian occupying x's next position. State_d checks if there is any wall between x and x's next position. The different state value stands for different states, as follows.

1) State_a = 1, $T \geq Tx$. State_a = 0, $T < Tx$.

4) State_b = 2, $Px \in D$. State_b = 1, $Px \in Sa$. State_b = 0, $Px \in Sb$.

5) State_c = 1, $Px \wedge$ Occupy (y). State_c = 0, $Px \wedge \neg$ Occupy (y).

6) State_d = 1, Between (Px, Px', Wpq).

   State_d = 0, $\neg$ Between (Px, Px', o).

3.1.3 Movement Rules

For every pedestrian, x, there are 8 rules to check his/her states and to decide his/her movement (distance and direction). Simulation program iteratively checks every x's state, and decides which rule is proper for x.

R1：State_a = 0 $\rightarrow$ Mx = 0

R2：State_a = 1 $\wedge$ State_b = 1 $\wedge$ State_c = 1 $\rightarrow$ Mx = 0

R3：State_a = 1 $\wedge$ State_b = 1 $\wedge$ State_c = 0 $\wedge$ R(x,a,b)

$\rightarrow$ $(Mx = t \cdot Vx) \wedge (Nx = D(x, b))$

R4：State_a = 1 $\wedge$ State_b = 0 $\wedge$ State_c = 1 $\rightarrow$ Mx = 0

R5：State_a = 1 $\wedge$ State_b = 0 $\wedge$ State_c = 0 $\wedge$ State_d = 0

$\rightarrow$ $(Mx = t \cdot Vx) \wedge (Nx = D(x, Dx))$

R6：State_a = 1 $\wedge$ State_b = 0 $\wedge$ State_c = 0 $\wedge$ State_d = 1

Ù Wpq Ù (L(p, Dx) ≦ L(q, Dx)) ® (Mx = t˙Vx) Ù (Nx = D(x, p))

R7：State_a = 1 Ù State_b = 0 Ù State_c = 0 Ù State_d = 1

Ù Wpq Ù (L(p, Dx) ≧ L(q, Dx)) ® (Mx = t˙Vx) Ù (Nx = D(x, q))

R8：State_b = 2 ® Mx = 0

3.2 Velocity Distribution

By observing pedestrians walking in several campus buildings, we analyse pedestrians' velocity distribution and get the following data, where there are 7 different speed intervals (V_1 -- V_7) with certain percentages. Most of pedestrians have the velocity between 1.27 to 1.45 m/sec.

$P(x, V_1)$ = P(x, 2.23m/sec) = 1 % , $P(x, V_2)$ = P(x, 1.69m/sec) = 10 %

$P(x, V_3)$ = P(x, 1.45m/sec) = 37 %, $P(x, V_4)$ = P(x, 1.27m/sec) = 30%

$P(x, V_5)$ = P(x, 1.13m/sec) = 13 % , $P(x, V_6)$ = P(x, 1.02m/sec) = 8 %

$P(x, V_7)$ = P(x, 0.92m/sec) = 1 %,

3.3 Start-Running-Time Distribution

It is impossible to request every pedestrian to evacuate at same time, because everyone takes different time to aware danger, respond, and start running. We assume the distribution of start-running-time as follows (unit: second).

$P(x, T_1)$ = P(x, Tx < 5) = 20 %, $P(x, T_2)$ = P(x, 5 £ Tx <10) = 60 %

$P(x, T_3)$ = P(x, 10 £ Tx < 15) = 20 %

The velocity distribution and start-running-time distribution may vary among buildings, people and time of the day.

3.4 Actions of Agent

The following is an explanation of an agent's various actions. At first, we can try a random distribution model which gives an agent two attributes, for example, Vx=1.2m/sec and Tx=2 sec. As for the other models used for modelling all the agents' various states and actions. Table 1, for instance, shows one agent's direction, moving distance, and rules of movement in every second. In this case, the agent takes 10 seconds and moving for 7.2 meters to run to his destination. Generally there are three kinds of space for movement on the agent's route: a room, a corridor, and a staircase. Before arriving at the destination, he can do several actions, including starting, running, waiting, walking along a wall, or arriving at the staircase. Every agent has his own actions, way to interact with other agents, and route to the staircase. That is, all agents' actions compose a dynamic process. This simulation process will end when the last agent arrives at the staircase. We use the computer to assist us to execute these models and to visualise the process.

Table 1 An Agent's Movement

T	Space	State_a	State_b	State_c	State_d	Applied Rule	Move Direction	Move Distance
1	Sa	0	1	0	0	R1		0
2	Sa	1	1	0	0	R3	D(x, b)	1.2
3	Sa	1	1	1	0	R2		0
4	Sa	1	1	0	0	R3	D(x, b)	1.2
5	Sb	1	0	0	0	R5	D(x, Dx)	1.2
6	Sb	1	0	0	1	R6	D(x, p)	1.2
7	Sb	1	0	1	0	R4		0
8	Sb	1	0	0	0	R5	D(x, Dx)	1.2
9	Sb	1	0	0	0	R5	D(x, Dx)	1.2
10	D	1	2	0	0	R8		0

4 CASE STUDY

A case study was conducted in a campus building in National Taiwan University. Figure 1 is the plan of the building, which displays layouts of rooms, corridors, staircases, etc. We label 5 staircases as "a", "b", "c", "d", "e". Small dots represent pedestrian in rooms. (Total: 235 people).

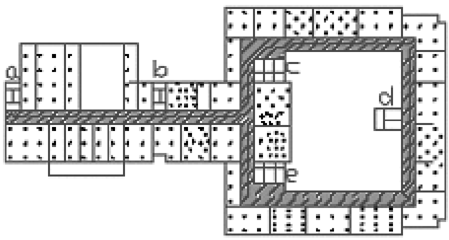

Figure 1 The floor plan of the case study building.

A simulation program is written by AutoLISP language and executed in an AutoCAD environment (Figure 2). We may locate agent with attributes, assign object positions, execute the activity rules, and visualise the dynamic process.

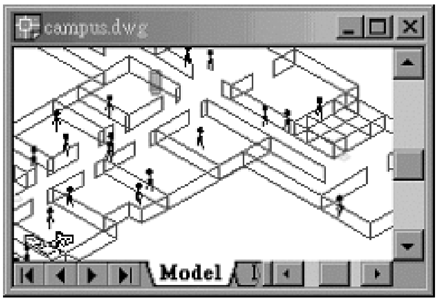

Figure 2 Simulation of agents, facilities, and paces.

Table 2 illustrates some important or serious combinations of staircase faults.

Table 2 Simulation Results

Available	Fail	K	Evacuation time		Pedestrian going to each staircase a	b	c	d	e	SD	DS
a,b,c,d,e		0	T	43	31	52	54	54	44	9.8	
b,c,d,e	A		T_a	64		83	54	54	44	16.8	7.0
a,c,d,e	B	1	T_b	62	44		78	54	59	14.3	4.5
a,b,d,e	C		T_c	59	31	56		74	74	20.4	10.6
a,b,c,e	D		T_d	92	31	52	91		61	24.9	15.1
a,b,c,d	E		T_e	69	67	30	51	87		24.1	14.3
c,d,e	a, b		T_{ab}	105			111	54	70	29.4	19.6
a,b,d	c, e	2	T_{ce}	74	31	112		92		42.2	32.4
a,d,e	b, c		T_{bc}	65	46			78	111	32.5	22.7
a,c,d	b, e		T_{be}	69	45		128	62		43.8	34.0
a,d	b,c,e	3	T_{bce}	105	83			152		48.8	39.0
a,b	c,d,e		T_{cde}	120	31	204				122.3	112.5
a	b,c,d,e	4	T_{bcde}	170	235						

4.1 Fault Tolerance Checking

4.1.1 FT^k Value

If it requires FT^k value are: FT^1 -- 90sec, FT^2 -- 135sec, FT^3 -- 180sec,

then k=1, Max $(T_a^1, T_b^1, T_c^1, T_d^1, T_e^1) = 92 > FT^1$, Not tolerable.

k=2. Max $(T^2_{ab}, T^2_{ce}) = 105 < FT^2$, Tolerable.

k=3, Max $(T^3_{bce}, T^3_{cde}) = 120 < FT^3$, Tolerable.

4.1.2 α^k Value

If it requires $\alpha^1 = 2$, $\alpha^2 = 3$, $\alpha^3 = 4$,

then $\forall\ i \in D$, $\frac{T_i}{T} < \alpha^1 = 2$, it is not true, $\frac{T_d}{T} = 2.13$, Not tolerable.

$\forall\ i, j \in D$, $\frac{T_{ij}}{T} < \alpha^2 = 3$, it is true, Tolerable.

$\forall\ i, j, k \in D$, $\frac{T_{ijk}}{T} < \alpha^3 = 4$, it is true, Tolerable.

4.1.3 NT Value

When all staircases work normally, N=5, $T^n = 43$. If it requires NT value is less than 5 x 43 = 215, then

k=1 , $T^1 \le 215/(5-1) = 53.75$ k=2 , $T^2 \le 215/(5-2) = 71.67$

k=3 , $T^3 \le 215/(5-3) = 107.5$

After checking NT value, only T_{bce} can fit requirement, but other conditions are not tolerable.

4.2 Fault Influence on Crowd Distribution

When some staircases fail, pedestrians may change their destinations, and the number of pedestrians toward each staircase may also change.

4.2.1 Discussion 1

1) For staircase a, when other staircase fails, β_e (69/31=2.23) is higher than β_b (44/31=1.4). It means b's fail has a serious impact on staircase a.
2) For staircase b, when other staircase fails, β_a (83/52=1.59) is highest value, which means staircase b has more pedestrians to digest.

3) For staircase c, d, and e, we may use same method to check each staircase.

4.2.2 Discussion 2

1) When k = 1, fault on d and e make SD increase obviously, which stands for serious situations.
2) When k increases, SD also increases, which means more faults will cause more unequal crowd, hence more dangerous situation.

5 CONCLUSION

This paper considers situations when some staircases are not able to function properly in emergency moments. Designers should assess the efficiency and the effectiveness of the remaining staircases and make sure building has sufficient staircase fault tolerance when real emergency happens. Our conclusions are as follows.

First, to effectively comply with performance-based-codes, computer simulations for analysing building's various performances are needed.

Second, a safe staircase-layout shouldn't mean it has good performance at normal condition only, but it should also have the ability to tolerate certain staircase fault.

Third, staircase fault tolerance could potentially be a new evaluation and should be taken into account when an architect designs spaces.

REFERENCES

Anderson, T. and P.A. Lee. 1981. *Fault Tolerance, Principles and Practice.* Englewood Cliffs: Prentice/ Hall International.

Jo, J.H. and J.S. Gero. 1998. Space Layout Planning using an Evolutionary Approach. *Artificial Intelligent Engineer,* vol. 12, 149-162.

Jonker, C.M. and J. Treur. 2001. Agent-based Simulation of Animal Behaviour. *AppliedIntelligence,* vol. 15, No.2, 83-115.

Karyagina, M.N. 1997. Designing for Fault-tolerance in the Commercial Environment. *Microelectronics and Reliability,* vol. 37, Issue: 4, 693.

Liggett, R.S. 2000. Automated Facilities Layout: Past, Present and Future. *Automation and Construction,* vol. 9, 197-215.

Templer, J. 1992. *The Staircase: Studies of Hazards, Falls, and Safer Design.* Cambridge: MIT Press.