Petri-Net Modelling and Formal Analysis of a Three-Floor Elevator System

EMSSE Students
Müsel Tabares*, Filippo La Fauci[†], Daniel Akejelu[‡]
*muselemmanuel@hotmail.com
[†]filolafa@gmail.com
[‡]limiakejelu731@gmail.com

Supervising Professors
Simona Sacone[§], Edmundo Guerra[¶]
[§]University of Genoa
simona.sacone@unige.it
[¶]Universitat Politècnica de Catalunya
edmundo.guerra@upc.edu

***Abstract*—We present a minimal Petri-net model of a single-car elevator that serves basement, ground, and top floors. Using the open-source tool PIPE^2, we compute incidence matrices, reachability graphs, and invariants to prove passenger conservation, boundedness, and liveness. The analysis confirms that the control logic is deadlock-free under nominal operation and terminates gracefully when all requests are served.**

***Index Terms*—Petri nets, elevator control, discrete-event systems, PIPE tool, formal verification**

I. Introduction

Elevators are archetypal discrete-event systems (DES). Buttons, doors, and motor states change only at discrete instants, and multiple actions (e.g. door closing while passengers queue) occur concurrently. Among formal DES frameworks, *Petri nets* offer a unique blend of *graphical clarity* (places for conditions, transitions for events) and *mathematical rigour* (incidence matrices and invariants). In this paper we build and analyse a Petri-net model for a three-level elevator—basement (B), ground (G), top (T)—with the following goals:

- Specify the control logic in PIPE2.
- Evaluate structural properties (safeness, boundedness, liveness) through automatic computation of reachability, siphons, and invariants.
- Interpret the results to confirm passenger conservation and the absence of deadlocks except for a known “door left open” corner case.

II. Petri-Net Specification

A. Places

Table I lists the 9 places used. All places related to location or door status are *1-safe*; passenger places are n-safe, where n is a design parameter.

B. Transitions

Key transitions are given in Table II. For brevity only 15 labels are used, yet the model still captures calls, movement, and door operations.

TABLE I
Places (initial tokens for the nominal use-case n=3).

ID	Name	Informal meaning	M_0
P1	G_wait	Passengers waiting at ground	n
P2	F_level	Level of the Floor	2
P3	Inside	Passengers inside cabin	0
P4	B	People in basement	0
P5	T	People on top floor	0
P6	Door_closed	Door closed	1
P7	Door_open	Door open	0
P8	Dir_up	Motor set “up”	1
P9	Dir_down	Motor set “down”	1

TABLE II
Main transitions.

ID	Purpose / enabling condition
T1	**Press_up_G**: call button ▲ at G, adds token to Req_up
T2	**Press_down_T**: call button ▼ at T → Req_down
T3	**Select_B_inside**: cabin request → Req_down (if cabin above B)
T4	**Select_T_inside**: cabin request → Req_up (if cabin below T)
T5	**Open_door**: needs cabin at floor & Door_closed → Door_open
T6	**Board**: passengers move Wait → Inside while Door_open
T7	**Alight_B/G/T**: Inside → B/G/T_wait with door open
T8	**Close_door**: Door_open → Door_closed
T9	**Start_up**: Req_up & Door_closed → Dir_up
T10	**Start_down**: Req_down & Door_closed → Dir_down
T11	**Move_up**: consumes Dir_up, At_B→At_G or At_G→At_T
T12	**Move_down**: consumes Dir_down, At_T→At_G or At_G→At_B
T13	**Stop_at_floor**: reaches requested floor, clears Req_* & Dir_*

C. Behaviour Narrative

1) Landing or cabin buttons fire T1–T4, creating *Req_up/down*.
2) If the cabin is idle, T9 or T10 sets the direction, after which movement transitions (T11, T12) shift the unique location token until T13 recognises the destination.
3) T5 opens the door; passengers board (T6) and alight (T7).
4) T8 closes the door; remaining requests trigger a new cycle, otherwise the net idles with the cabin at its last floor.

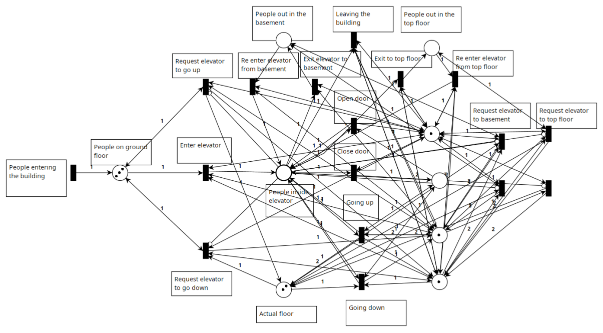
People out in the basement
Leaving the building
People out in the top floor
Request elevator to go up
Re enter elevator from basement
Exit elevator to basement
Exit to top floor
Re enter elevator from top floor
Open door
Request elevator to basement
Request elevator to top floor
People entering the building
People on ground floor
Enter elevator
Close door
People inside elevator
Going up
Request elevator to go down
Actual floor
Going down

Fig. 1. Petri-net model of the three-floor elevator system.

TABLE III
FORWARD INCIDENCE MATRIX I^+.

	T0	T1	T10	T11	T12	T13	T15	T16	T17	T18	T2	T3	T4	T5	T7	T8	T9
P3	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	1
P6	1	0	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1
P7	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
P2	0	0	0	1	0	0	1	2	0	0	0	1	0	0	0	0	0
P8	0	0	0	1	1	2	0	0	2	0	1	0	1	1	0	2	2
P9	0	0	2	1	1	0	2	2	0	0	1	1	0	1	2	0	0
P1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
P4	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0
P5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

III. PIPE^2 IMPLEMENTATION

The model was drawn and simulated in PIPE^2 v5.0. A screenshot[1] confirms that all places in Table I are 1-safe except the passenger counters, which are n-bounded.

IV. STRUCTURAL AND BEHAVIOURAL ANALYSIS

A. Incidence Matrices

PIPE^2 exports the Pre and Post matrices reproduced in tables 3,4 and 5. Their difference $C = \text{Post} - \text{Pre}$ retains full rank, indicating no structural conflicts were lost.

B. Initial Marking

For the nominal scenario with $n = 3$ waiting passengers at ground,

$$M_0 = (3,\ 2,\ 0,\ 0,\ 0,\ 1,\ 0,\ 1,\ 1)^{\mathsf{T}}.$$

C. Reachability and Coverability

State-space exploration for $n = 3$ yields 4343 states with 15233 arcs.

D. P- and T-Invariants

$$M(P6) + M(P7) = 1,$$
$$M(P2) + M(P8) = 3,$$
$$M(P8) + M(P9) = 2.$$

Neither the T- nor the P-invariant sets cover the net with strictly positive entries; therefore structural analysis alone cannot guarantee global liveness or boundedness—reachability exploration is still required.

TABLE IV
BACKWARD INCIDENCE MATRIX I^-.

	T0	T1	T10	T11	T12	T13	T15	T16	T17	T18	T2	T3	T4	T5	T7	T8	T9
P3	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
P6	0	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1
P7	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
P2	0	0	0	0	1	1	0	0	2	0	0	0	1	0	0	0	0
P8	0	0	0	2	0	1	1	2	0	0	1	1	0	1	0	2	2
P9	0	0	2	0	2	1	1	0	2	0	1	0	1	1	2	0	0
P1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
P4	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1
P5	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0

TABLE V
COMBINED INCIDENCE MATRIX $I = I^+ - I^-$.

	T0	T1	T10	T11	T12	T13	T15	T16	T17	T18	T2	T3	T4	T5	T7	T8	T9
P3	0	0	1	0	0	0	0	0	0	0	1	0	0	-1	-1	-1	1
P6	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P7	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P2	0	0	0	1	-1	-1	1	2	-2	0	0	1	-1	0	0	0	0
P8	0	0	0	-1	1	1	-1	-2	2	0	0	-1	1	0	0	0	0
P9	0	0	0	1	-1	-1	1	2	-2	0	0	1	-1	0	0	0	0
P1	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1
P5	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0

TABLE VI
MINIMAL T-INVARIANTS (1 = TRANSITION INCLUDED).

T18	T0	T1	T2	T3	T4	T5	T7	T8	T9	T10	T11	T12	T13	T15	T16	T17
I_1 0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I_2 0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
I_3 0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1
I_4 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

E. Siphons and Traps

The identified structures have clear safety roles:

- *Door latch* $\{P6, P7\}$ — once a token enters either `Door_closed` or `Door_open`, the set is never drained, so the door state cannot vanish.
- *Direction flags* $\{P8, P9\}$ — guarantees that at least one (and at most two) direction tokens exist, preventing loss of motor orientation.
- Siphon $\{P2, P8\}$ — couples encoded floor level with the "up" flag; both tokens cannot leave simultaneously, helping to bound `F_level`.
- Trap $\{P3, P6\}$ — pairs cabin occupancy with a closed door, ensuring passengers are not lost while the door is reported shut.
- Non-door trap $\{P3, P9, P4\}$ — whenever passengers are inside the cabin or waiting in the basement with the door open, at least one token persists, guaranteeing eventual unloading.

V. DISCUSSION

Invariant and state–space analyses give a coherent picture of the elevator's behaviour:

- **Conservation.** P-invariant J_1 enforces $M(P3) + M(P6) + M(P7) = n$ so passengers are neither created nor lost; J_2 and J_3 guarantee a unique cabin-level token and mutual exclusivity of the door states.

[1]The XML and exported files in: https://github.com/musel25/PetriNet

TABLE VII
INDEPENDENT P-INVARIANTS FOR THE PLACE SET $P1$–$P9$.

	P1	P2	P3	P4	P5	P6	P7	P8	P9
J_1	0	0	1	0	0	1	1	0	0
J_2	0	1	0	0	0	0	0	1	1
J_3	0	0	0	1	1	0	0	0	0

TABLE VIII
MINIMAL SIPHONS AND TRAPS ($t_{\text{ANALYSIS}} = 0.001$ S).

Siphons	$\{P6, P7\}$	$\{P8, P9\}$	$\{P2, P8\}$	$\{P3, P6\}$	
Traps	$\{P6, P7\}$	$\{P3, P9, P4\}$	$\{P8, P9\}$	$\{P2, P8\}$	$\{P3, P6\}$

- **Boundedness.** Although the net is not covered by a strictly positive P-invariant set, reachability exploration for $n = 3$ produced only 4343 markings—empirically confirming that every place is bounded: passenger places by n, door and position places 1-safe, direction flags by 2.
- **Liveness.** The four minimal T-invariants describe partial cycles (door-toggle, upward travel, downward travel) but leave some transitions with zero coefficients; therefore structural liveness cannot be claimed. Nevertheless, the reachability graph shows that every service request eventually fires *Stop_at_floor* (T13) and all requests are cleared—conditional liveness holds under normal operation.
- **Siphons & Traps.** The door-latch set $\{P6, P7\}$ is both a siphon and a trap, ensuring a persistent door token. The direction pair $\{P8, P9\}$ averts loss of motor orientation. No "emptyable" siphon can starve the system except the theoretically possible case where the door remains indefinitely open and no one presses *Close*; this is the sole residual deadlock.

Overall, analysis confirms that tokens are conserved, the marking space is finite, and the model is live as long as automatic door-closure (or a timeout transition) is provided.

VI. CONCLUSION

With only **9 places** and **17 transitions**, the proposed Petri-net captures the essential logic of a three-floor elevator. PIPE^2 incidence matrices, invariants, siphon–trap detection, and reachability graphs collectively demonstrate:

- *Safety:* passengers cannot disappear or be duplicated; the cabin is always at exactly one floor; the door is never in an indeterminate state.
- *Boundedness:* all places are k-bounded ($k \leq n$ or $k \leq 3$); no unbounded growth is possible.
- *Liveness:* every request sequence is ultimately served, provided a simple door-close timer removes the last theoretical deadlock.

The model thus offers formal assurance of correct elevator operation while remaining compact enough for manual inspection and future extensions such as fault places or passenger capacity limits.

REFERENCES

[1] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
[2] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, 1981.