Elevator Controller

Software Architecture Design

SAD Version 2.0

Team #7
13 November 2025

Youssef Amin
Valerie Barker (manager)
Natalie Runyan
Daniel Thompson
Joel Villarreal

CS 460 Software Engineering

Table of Contents

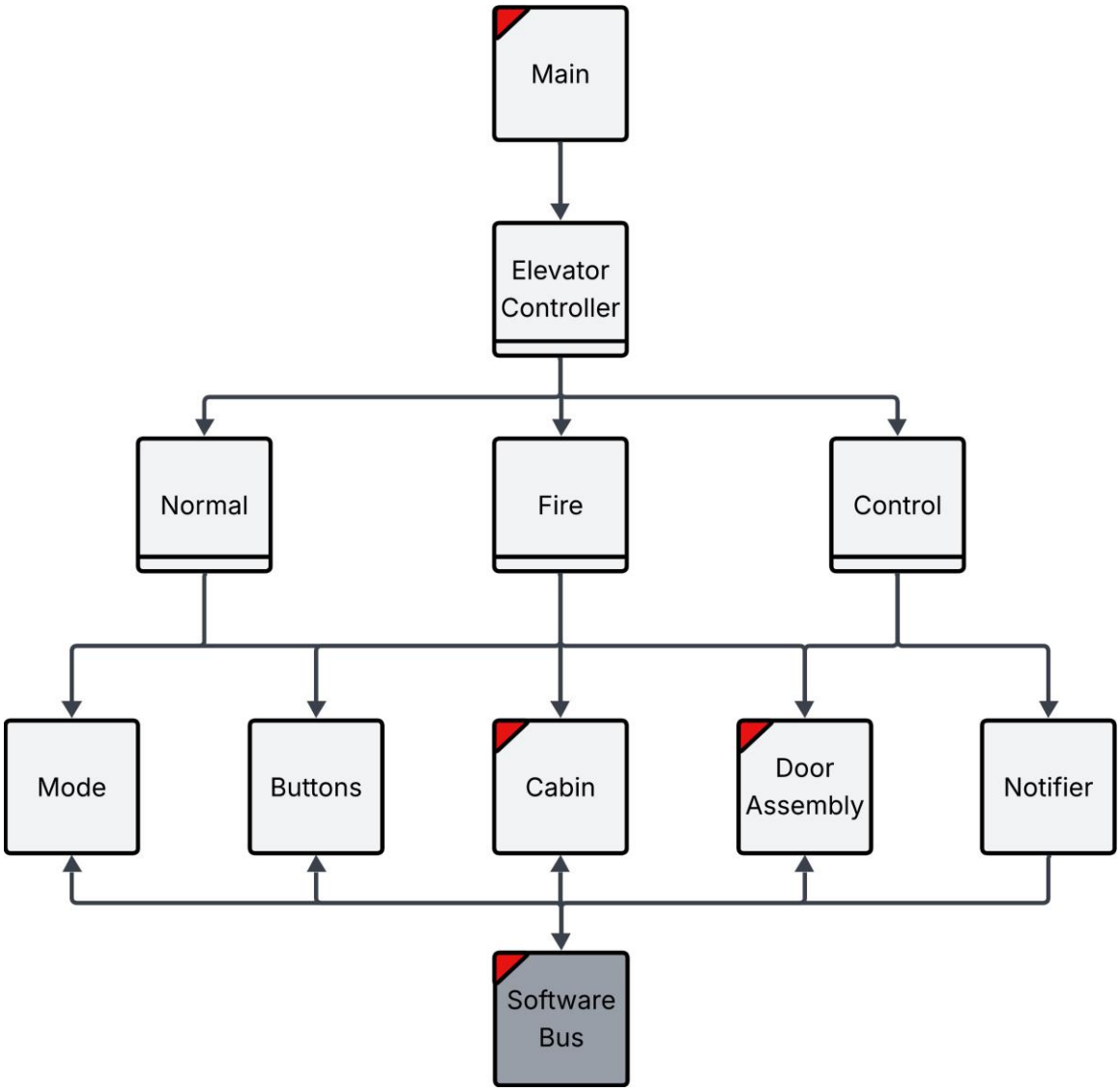
1 Introduction

The Elevator Control System governs the physical and logical behavior of the elevator. Managing motion, ensuring safety, and receiving user input. It integrates motion control with user-facing devices through a software bus. The core components are as follows:

Normal, which is the default mode that the elevator control system is started in.
Fire, the mode when there is an emergency, sends all the elevators to the ground floor and only listens to button presses when the fire key is inserted.
Control, mode where the elevator listens to commands from the supervisor.
Mode, which handles mode changes such as on\off\fire-safety modes.
Button-Presses, which contains all the request button presses in the cabin, as well as call button presses on levels.
Cabin, which handles the elevator's movement.
Door Assembly, which handles operations related to opening and closing the doors and informs control when the elevator has reached the weight capacity. It also handles arrival noises in the speakers.
Notifier, that is responsible ensuring both audio and visuals aptly describe the state of the elevator.

These interfaces communicate with the appropriate hardware and software devices in order to ensure proper elevator function. They each contain operations for communications via a subscription-based software bus pattern. Programs may use the API to access information about hardware components in order to ensure safe, reliable behavior.

2 Design Diagram



The Design Diagram above demonstrates the structure of the Elevator Controller sub-system. The Elevator Controller interacts directly with the Mode, Buttons, Cabin, Door Assembly, and Notifier via their API in order to support the full functionality of an elevator within the larger system. Notice that all of the objects that interface with the Software Bus receive messages, except for the Notifier which only publishes messages. Additionally, the Mode and Buttons objects are the only objects that do not publish messages to the software bus. All the lower-level objects are abstractions of hardware and software, meaning the software Bus is a level of indirection between our system and the broader system, allowing the system as a whole to maintain maximal mobility. Since the software bus is an outside entity, its software architecture is not specified in this document.

3 Component Specifications

3.1 Main

Main is a lightweight object, which instantiates Elevator Controller, Mode, Buttons, Cabin, Door Assembly and Notifier.

3.2 Elevator Controller

The elevator controller is a light-weight object responsible for switching between the elevator's various modes.

void elevatorController()

3.3 Normal

Normal mode is the default mode that the system starts in. The initial state of each elevator is on the first floor with the doors open. In this mode, no messages from the supervisor are expected, other than a change in mode; the movement is determined solely by button presses. This mode provides the typical elevator functionality: handling requests and navigating floors.

Mode normal()

3.4 Fire

In fire mode, the elevator only listens to request buttons in the cabin if the fire key has been inserted. Only one service button can be lit up at a time. If two buttons are pressed, the most recently pressed button is the only service request.

Mode fire()

3.5 Control

Control mode is one where movement is controlled by the Control Room. Using the Software Bus. The Control Room can give commands to the elevators and assumes full control over the system.

Mode control()

3.6 Mode

The mode serves as a means for the Elevator Controller to be put into and track its current mode. The mode is indirectly being updated by the Control Room, a separate entity outside of the Elevator Controller system. Additionally, the mode is responsible for taking in demands from the Control Room when the elevator is being remotely controlled. The mode object receives messages via the software bus but does not post messages to the software bus.

The modes:

- 1 – NORMAL
- 2 – FIRE_SAFETY

- 3 - CONTROLLED

Mode getMode()
(floor,direction) nextService()

3.7 Buttons

The buttons object enables the Elevator Controller to track and schedule its destinations. The buttons object indirectly receives floor requests via the physical buttons on the panel inside of the cabin, as well as the call buttons on each level. These button events are being received via the software bus. The buttons object does not post any messages to the Software Bus.

void callReset(floor,direction)
void requestReset(floor)
void enableCalls()
void disableCalls()
void enableAllRequests()
void enableSingleRequest()
(floor,direction) nextService(floor,direction)

3.8 Cabin

The cabin provides a means for the elevator controller to send the elevator to a destination. A note on the behavior of setting the destination of the cabin: once the cabin is fully stopped it can be sent in any direction, but if it is already in motion, the cabin will only obey requests to go to floors that are in the direction it is currently traveling. The cabin indirectly controls the motor by sending messages to the Software Bus. Additionally, the cabin indirectly receives messages from physical sensors through the Software Bus.

void goToFloor(floor)
(floor,direction) currentStatus()
boolean arrived()
floor getTargetFloor()

3.9 Door Assembly

The door assembly is a virtualization of the physical interfaces which comprise the doors: fully open sensors, fully closed sensors, door obstruction sensors, the scale, and the door motor. The door assembly posts and receives messages from its physical counterparts via the software bus; posting to the motor; receiving from the fully closed sensors, fully open sensors, the scale, and the door obstruction sensors.

void open()
void close()
boolean obstructed()
boolean fullyClosed()
boolean fullyOpen()

boolean overCapacity()

3.10 Notifier

The notifier object is used to communicate all necessary visual and audio information. The notifier sends messages to the speakers, button lights, and floor display (up/down arrows and LEDs for displaying the floor number). The notifier object does not receive any messages from the Software Bus.

void arrivedAtFloor(floor, direction)
void elevatorStatus (floor, direction)
void playCapacityNoise()
void stopCapacityNoise()

4 Design Constraints

1. Communication with hardware must be implemented with a software bus
2. Code must be implemented in Java
3. Code must be modular: simulated hardware easily replaced
5. Ensure the elevator has proper alignment with floors
7. Ensure the elevator reaches destinations in a timely fashion
8. Supports multiple modes: Fire-Safety, Normal, and Off.
6. Ensure functionality of API

5 Definition of Terms

- API: Application Programming Interface
- Cabin: the place in the elevator system where passengers wait while being transported. Also colloquially known as the elevator car or elevator cab.