A product is to be installed to control n elevators in a building with m floors. The problem concerns the logic required to move elevators between floors according to the following constraints:

C_1. Each elevator has a set of m buttons, one for each floor. These illuminate when pressed and cause the elevator to visit the corresponding floor. The illumination is canceled when the corresponding floor is visited by the elevator.

C_2. Each floor, except the first floor and top floor, has two buttons, one to request an up-elevator and one to request a down-elevator. These buttons illuminate when pressed. The illumination is canceled when an elevator visits the floor and then moves in the desired direction.

C_3. When an elevator has no requests, it remains at its current floor with its doors closed.

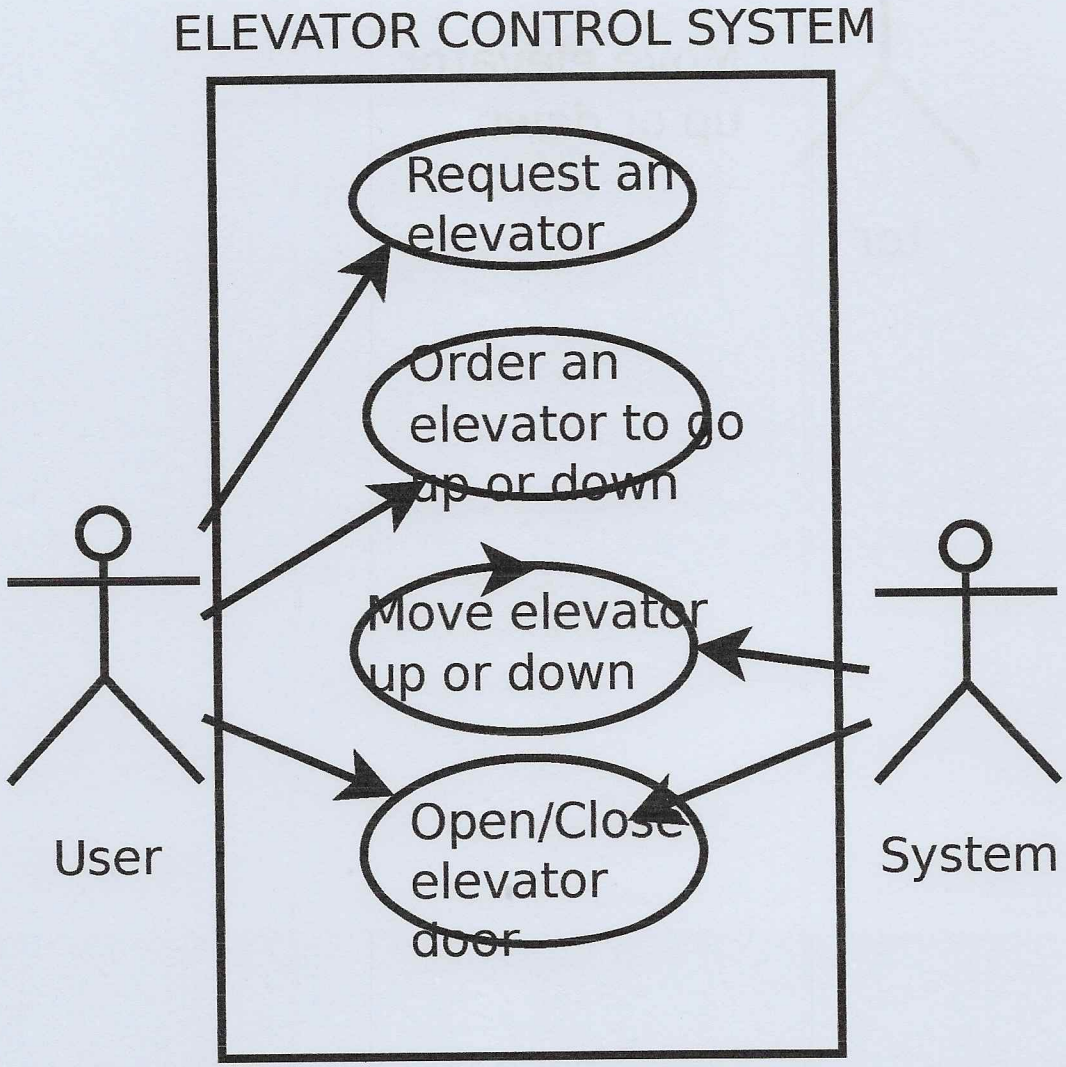
ELEVATOR CONTROL SYSTEM
Request an elevator
Order an elevator to go up or down
Move elevator up or down
Open/Close elevator door
User
System

Figure 12.11
Normal scenario of Figure 11.2.

1. User A presses Up floor button at floor 3 to request elevator. User A wishes to go to floor 7.
2. Up floor button is turned on.
3. An elevator arrives at floor 3. It contains User B who has entered the elevator at floor 1 and pressed the elevator button for floor 9.
4. Up floor button is turned off.
5. Elevator doors open.

   User A enters elevator.
6. User A presses elevator button for floor 7.
7. Floor 7 elevator button is turned on.
8. Elevator doors close.
9. Elevator travels to floor 7.
10. Floor 7 elevator button is turned off.
11. Elevator doors open to allow User A to exit elevator.
12. Timer starts.

    User A exits.
13. Elevator doors close after timeout.
14. Elevator proceeds to floor 9 with User B.

Figure 11.4
First iteration of class diagram.

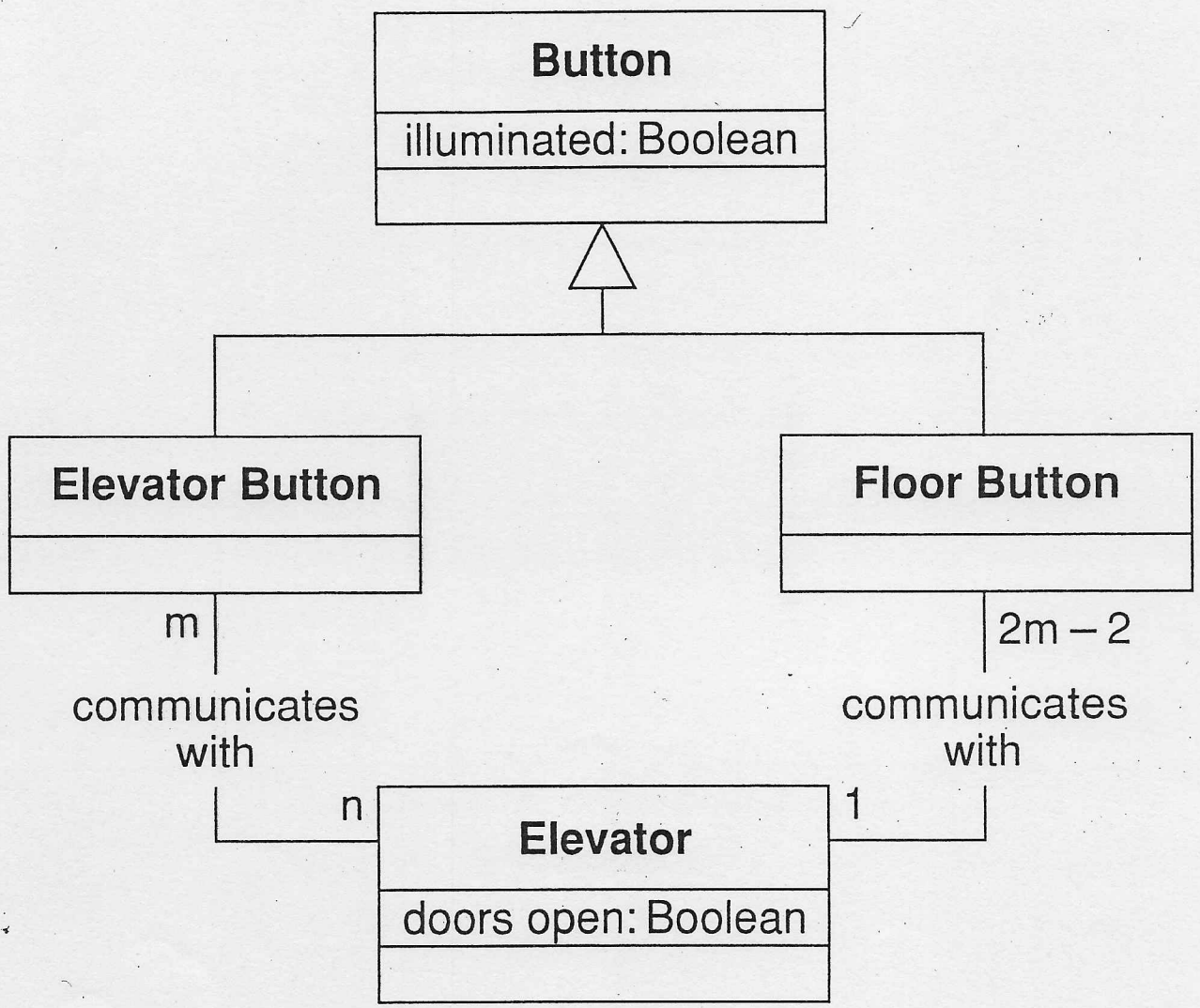

Figure 11.5
Second iteration of class diagram.

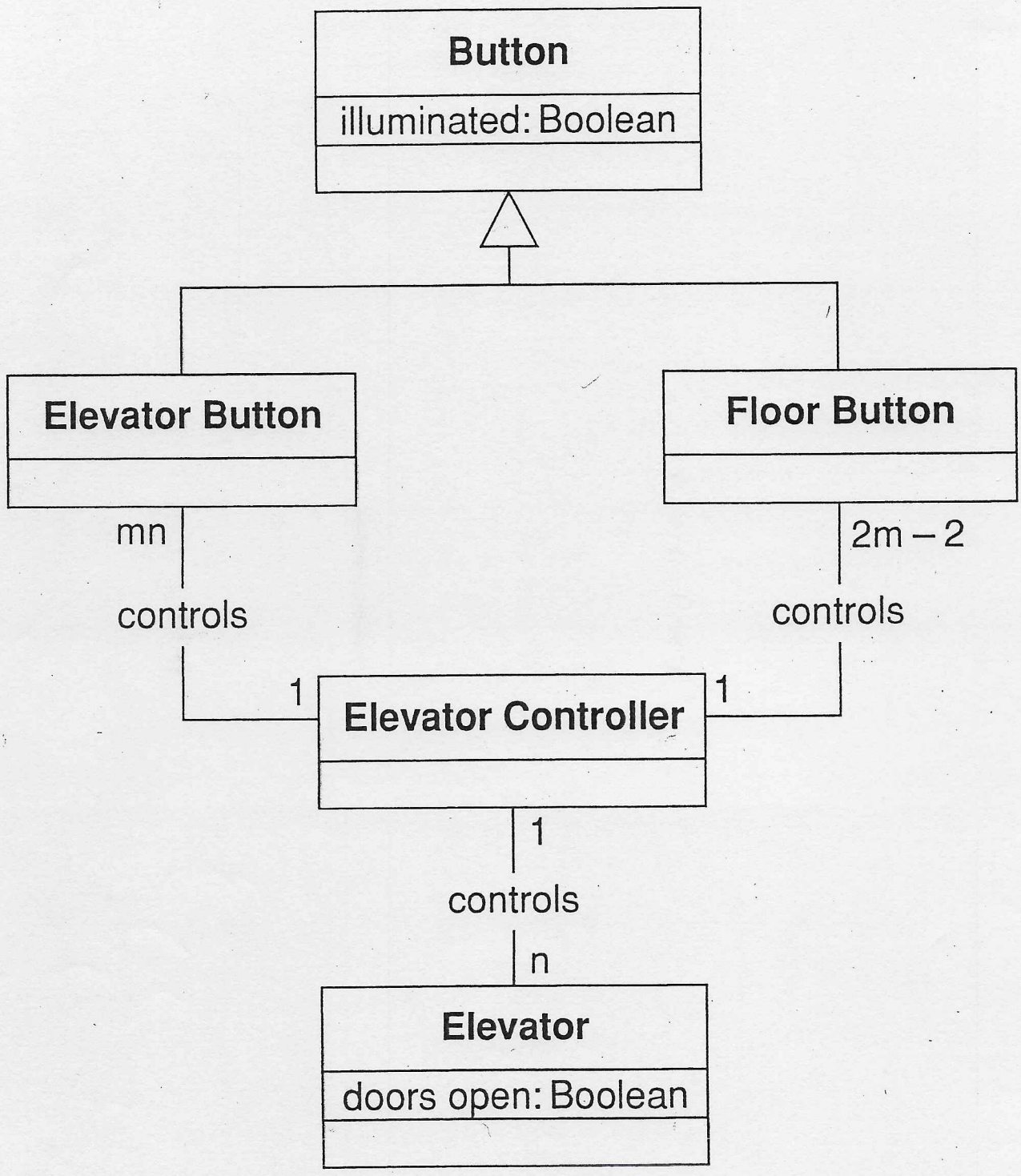

Figure 11.8
Third iteration of class diagram.

- where Elevator Application will be required for method main.

- and Elevator Utilities various utility routines will almost be certainly needed such as, timer, shortest distance algorithm. (remember O/S for disk controller)

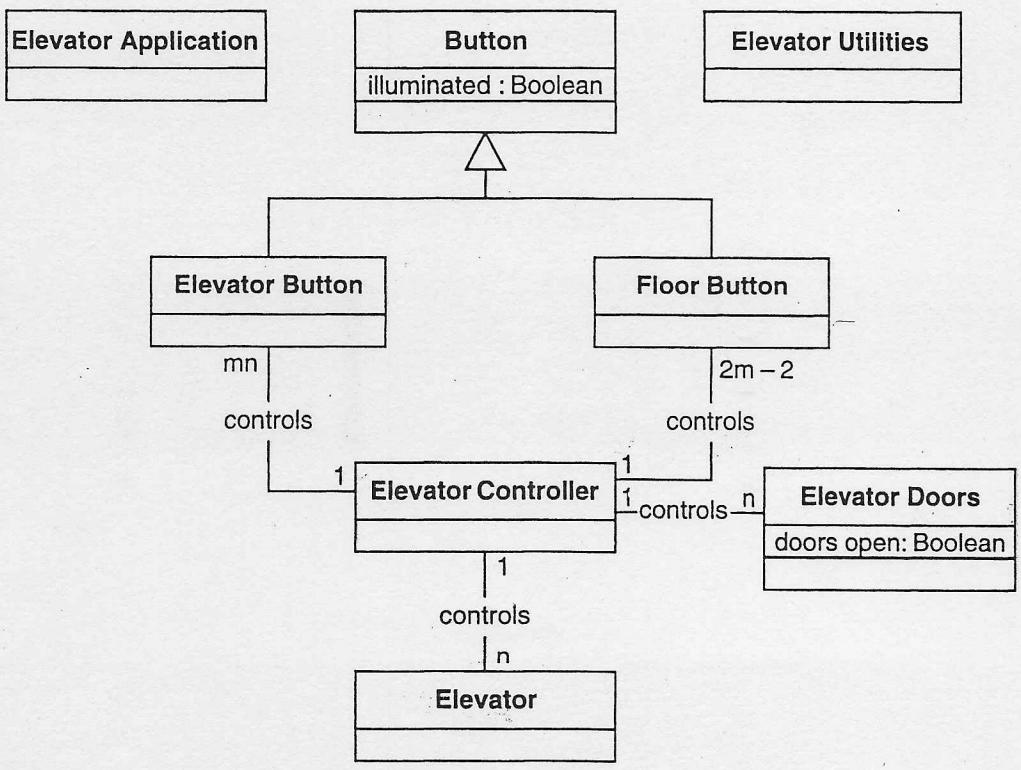

Figure 12.12
Sequence diagram for scenario of Figure 12.11.

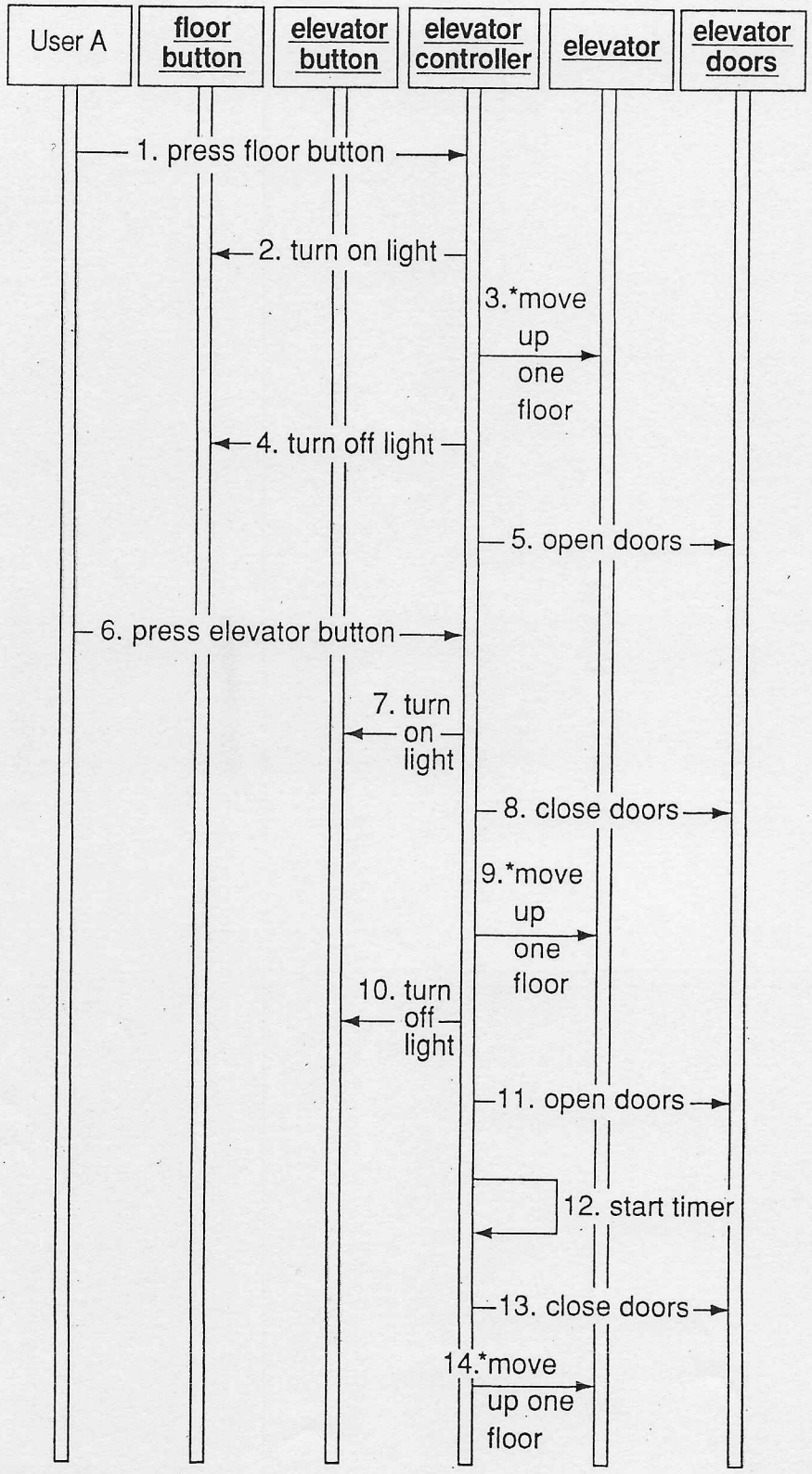

Figure 12.14
Detailed class diagram. (Additional state variables and methods will be added later to Elevator Controller.)

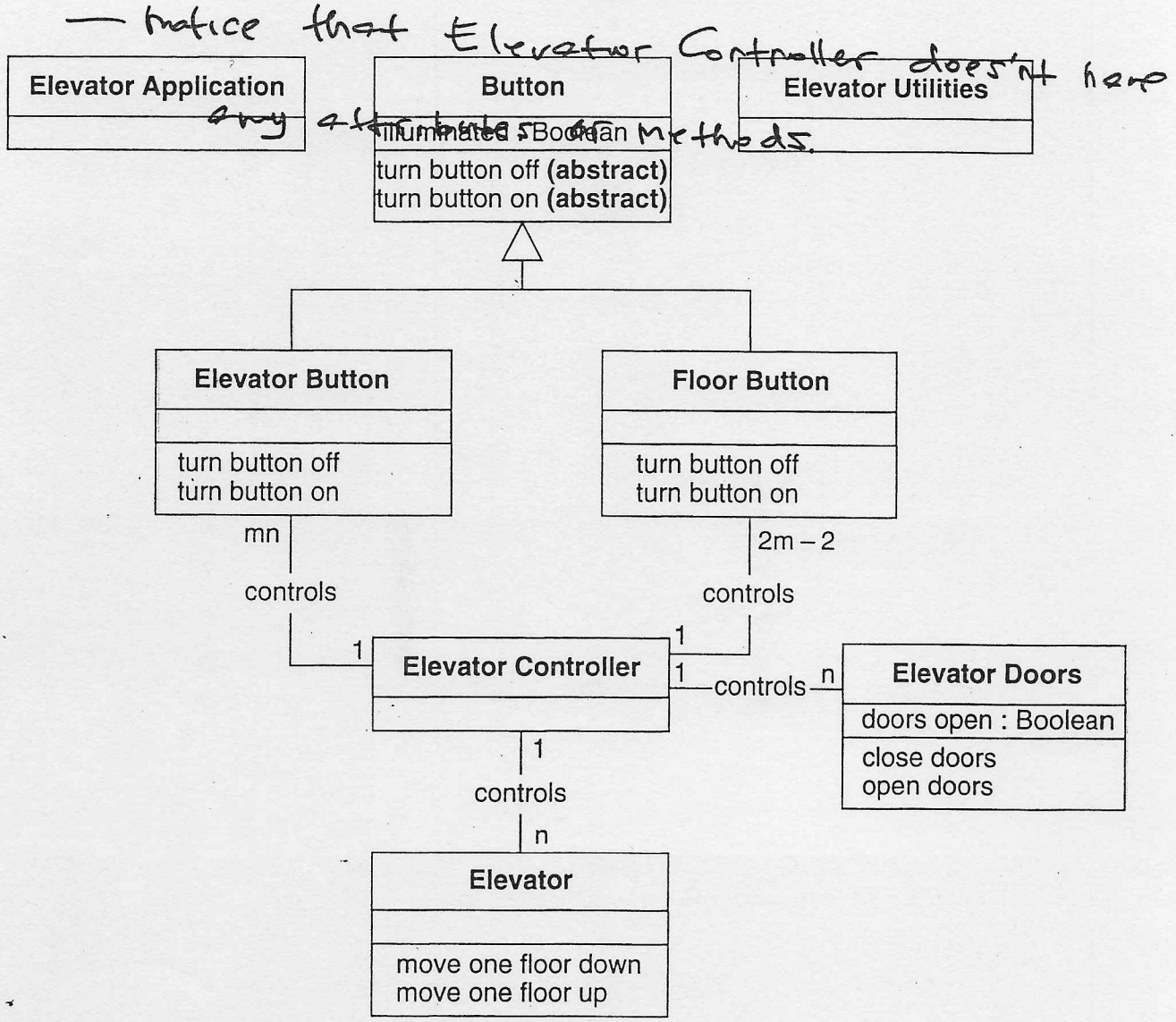

Figure 12.16
Detailed design of method elevator control loop.

```
void elevator control loop (void)
{
  do
  {
    if (a button has been pressed)
      if (button is not on)
      {
        button.turn button on;
        log request;
      }
    else if (elevator is moving up)
    {
      if (there is no request to stop at floor f)
        elevator.move one floor up;
      else
      {
        stop elevator by not sending a message to move;
        elevator doors.open doors;
        if (elevator button is on)
          elevator button.turn button off;
        update requests;
      }
    }
    else if (elevator is moving down)
      [similar to up case]
    else if (elevator is stopped and request is pending)
    {
      elevator doors.close doors;
      if (floor button is on)
        floor button.turn button off;
      determine direction of next request;
      elevator.move one floor up/down;
    }
    else if (elevator is at rest and not (request is pending))
      elevator doors.close doors;
    else
      there are no requests, elevator is stopped with elevator doors closed, so do nothing;
  }
}
```

Figure 11.3
An abnormal scenario.

1. User A presses Up fl oor button at fl oor 3 to request elevator. User A wishes to go to floor 1.
2. Up floor button is turned on.
3. An elevator arrives at floor 3. It contains User B who has entered the elevator at floor 1 and pressed the elevator button for floor 9.
4. Up floor button is turned off.
5. Elevator doors open.

   User A enters elevator.
6. User A presses elevator button for floor 1.
7. Floor 1 elevator button is turned on.
8. Elevator doors close after timeout.
9. Elevator travels to floor 9.
10. Floor 9 elevator button is turned off.
11. Elevator doors open to allow User B to exit elevator.
12. Timer starts.

    User B exits.
13. Elevator doors close after timeout.
14. Elevator proceeds to floor 1 with User A.