Northeastern University

软件工程

张爽

东北大学软件学院

5.5 Testing during OOA

CRC Cards

- **CRC**

  Class-Responsibility-Collaboration

CRC Cards

- **Weakness**
  - **Domain expertise is needed for OOA modeling**
- **Strength**
  - **Powerful tool for highlighting missing or incorrect items ---- for testing**

CRC Cards

- For each class, fill in a card showing
  - Name of Class
  - Functionality (Responsibility)
  - List of classes it invokes (Collaboration)

Testing during OOA

CLASS **Elevator Controller**
RESPONSIBILITY 1. Turn on elevator button 2. Turn off elevator button 3. Turn on floor button 4. Turn off floor button 5. Move elevator up one floor 6. Move elevator down one floor 7. Open elevator doors and start timer 8. Close elevator doors after timeout 9. Check requests 10. Update requests
COLLABORATION 1. Class **Elevator Button** 2. Class **Floor Button** 3. Class **Elevator**

CRC Cards

- **Consider responsibility**
  - *1. Turn on elevator button*
- **Problem**
  - **Totally unacceptable for object-oriented paradigm**
  - **Responsibility-driven design ignored**
  - **Information hiding ignored**
- **Responsibility**

  1. Turn on elevator button

  should be

  1. Send message to ElevatorButton to turn itself on

CRC Cards

- **A class has been overlooked**
  - ***ElevatorDoors* have a *state* that changes during execution (class characteristic)**
  - **Add class *ElevatorDoor***

CLASS

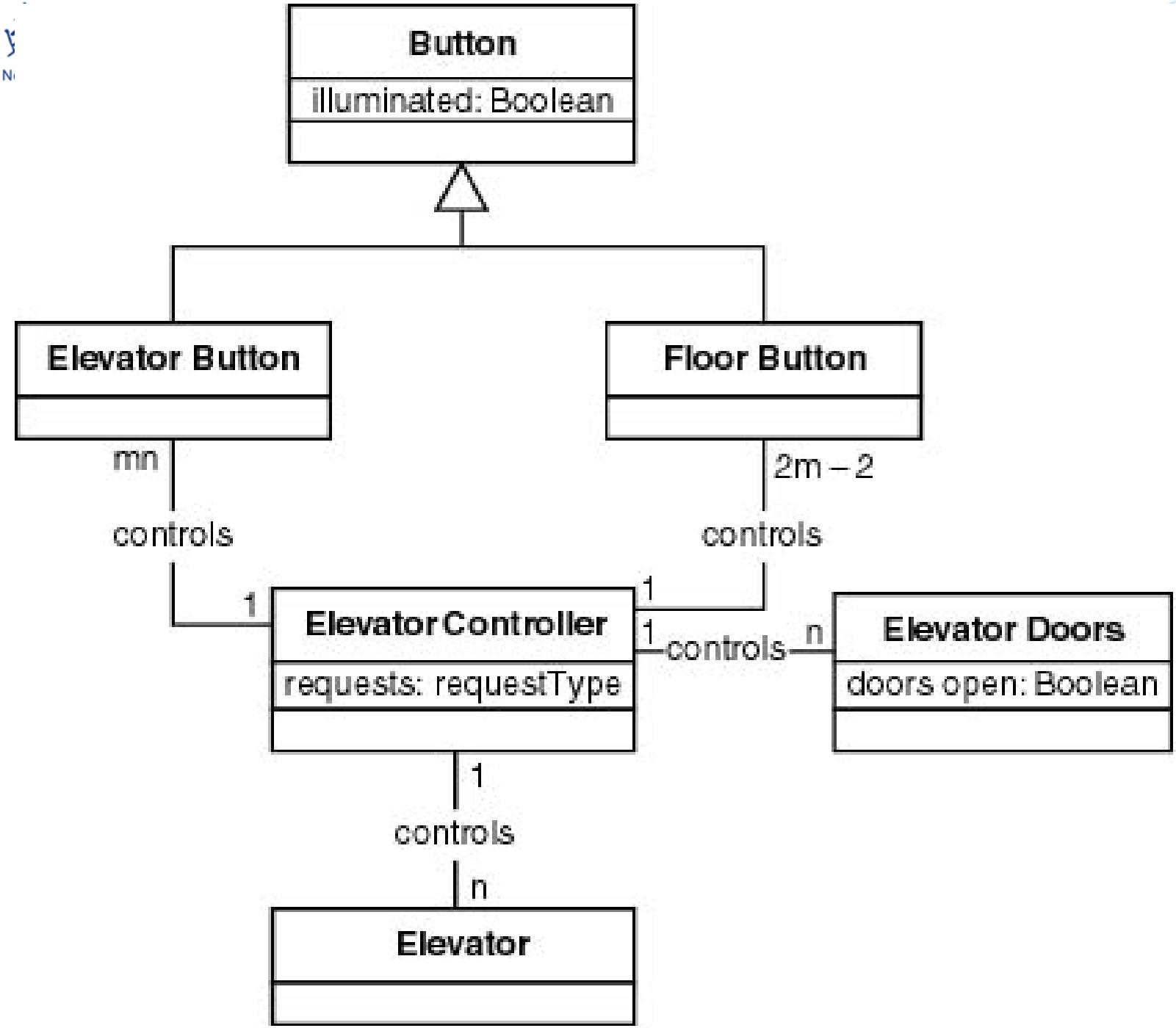
Elevator Controller

RESPONSIBILITY

1. Send message to **Elevator Button** to turn on button
2. Send message to **Elevator Button** to turn off button
3. Send message to **Floor Button** to turn on button
4. Send message to **Floor Button** to turn off button
5. Send message to **Elevator** to move up one floor
6. Send message to **Elevator** to move down one floor
7. Send message to **Elevator Doors** to open
8. Start timer
9. Send message to **Elevator Doors** to close after timeout
10. Check requests
11. Update requests

COLLABORATION

1. Subclass **Elevator Button**
2. Subclass **Floor Button**
3. Class **Elevator Doors**
4. Class **Elevator**


Button
illuminated: Boolean
Elevator Button
Floor Button
mn
2m – 2
controls
controls
1
1
Elevator Controller
requests: requestType
1
controls
n
Elevator Doors
doors open: Boolean
1
controls
n
Elevator

Elevator Problem: OOA

- Now, we have got correct class diagram for elevator system.
- So, the use case diagram and state diagram must be re-examined to see if they need further refinement.
- The use case diagram clearly still is adequate.
- The set of state diagrams must be extended to include the additional class **Elevator Door**.

Elevator Problem: OOA

- All three models are now fine.
- We should rather say:
  - All three models are fine *for now* .
- We may need to return to the object-oriented analysis phase during the object-oriented design phase.