

Object-Oriented Analysis Phase

- Specification phase for Object-Oriented paradigm
- Semiformal Technique
- Natural part of OOA is the graphical notation associated with the technique
- Learning to use OOA has Means learning the graphical notations for that technique

Object-Oriented Analysis

- OOA consists of three steps
 - Use-case modeling
 - Class modeling
 - Dynamic modeling

Object-Oriented Analysis

- Use-case modeling
 - Disregarding the sequence determines how the various products are computed
 - Display these info in use-case diagrams and associated scenarios
 - Scenarios are use-case instances
 - Action Oriented
 - Referred to as Functional Modeling

Object-Oriented Analysis

- Class modeling
 - Determine
 - Classes
 - Their attributes
 - The relationships between classes
 - Present information in class diagram

Object-Oriented Analysis

- Dynamic modeling
 - Determine actions performed by or to each class or subclass
 - Present information in a state diagram
 - Action oriented

Object-Oriented Analysis

- These three steps are not performed in sequence
 - Steps are performed in parallel
 - A change to a diagram will trigger changes to others
 - Diagrams are updated continuously
- The knowledge gained in OOA process is represented in different ways
 - Represents different aspects of the target product

Object-Oriented Analysis

- By the end of the process, the combined views/diagrams represent an overall understanding of the product

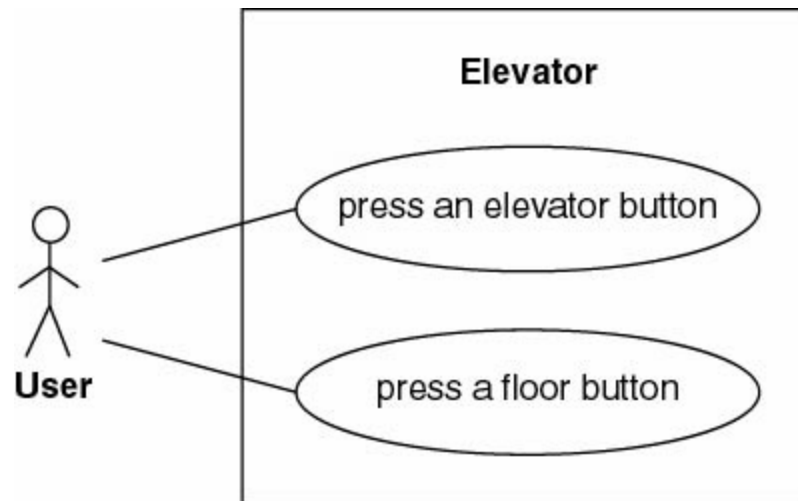
Elevator Problem

- A product that controls n elevators in a m story building
 - 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
 - Each floor, except the first 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
 - When an elevator has no request, it remains at its current floor with its doors closed

Use-Case Modeling

- Interactions possible between user and elevator
 - User pressing an elevator button to summon an elevator to that floor
 - User pressing a floor button requesting the elevator to stop at a specific floor

Use-Case Modeling: Use-Case Diagram



Use-Case Modeling: Normal Scenario

1. User A presses the Up floor button at floor 3 to request an elevator. User A wishes to go to floor 7.
2. The 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. The Up floor button is turned off.
5. The elevator doors open.
6. The timer starts.
User A enters the elevator.
7. User A presses the elevator button for floor 7.
8. The elevator button for floor 7 is turned on.
9. The elevator doors close after a timeout.
10. The elevator travels to floor 7.
11. The elevator button for floor 7 is turned off.
12. The elevator doors open to allow User A to exit from the elevator.
13. The timer starts.
User A exits from the elevator.
14. The elevator doors close after a timeout.
15. The elevator proceeds to floor 9 with User B.

Use-Case Modeling: An Exception Scenario

1. User A presses the Up floor button at floor 3 to request an elevator. User A wishes to go to floor 1.
2. The 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. The Up floor button is turned off.
5. The elevator doors open.
6. The timer starts.
User A enters the elevator.
7. User A presses the elevator button for floor 1.
8. The elevator button for floor 1 is turned on.
9. The elevator doors close after a timeout.
10. The elevator travels to floor 9.
11. The elevator button for floor 9 is turned off.
12. The elevator doors open to allow User B to exit from the elevator.
13. The timer starts.
User B exits from the elevator.
14. The elevator doors close after a timeout.
15. The elevator proceeds to floor 1 with User A.

Class Modeling

- Classes and their attributes are extracted and represented in a entity-relationship diagram
 - Only attributes of the class are determined not the methods
 - Methods are determined during Object-Oriented Design phase
- It is vary difficult to extract the classes and their attributes from problem statements or scenarios

Class Modeling

- Methods to extract classes
 - For developers with domain expertise
 - Deduce them from the use-cases
 - CRC Cards
 - For developers without domain expertise
 - Noun Extraction

Deduce them from the use-cases

- From the scenarios in Figures 12.2 and 12.3 candidate classes are
 - Elevator buttons
 - Floor buttons
 - Elevators
 - Doors
 - Timers

Noun Extraction

- 1) Concise Problem Definition
 - Define the product briefly, preferably in single sentences
 - Example:
 - Buttons in elevators and on the floors control the motion of n elevators in a building with m floors

Noun Extraction

- 2) Informal Strategy
 - Take the problem constraint into account
 - Example:
 - Buttons in elevators and on the floors control the motion of n elevators in a building with m floors. Buttons illuminate when pressed to request the elevator to stop at a specific floor; the illumination is canceled when the request has been satisfied. When an elevator has no requests, it remains at its current floor with its doors closed.

Noun Extraction

- 3) Formalize the Strategy
 - Identify the nouns in the informal strategy
 - Excludes nouns that lie outside the problem boundary
 - Use the nouns as candidate classes
 - Example:
 - Buttons in elevators and on the floors control the movement of n elevators in a building with m floors. Buttons illuminate when pressed to request the elevator to stop at a specific floor; the illumination is canceled when the request has been satisfied. When an elevator has no requests, it remains at its current floor with its doors closed.

Noun Extraction

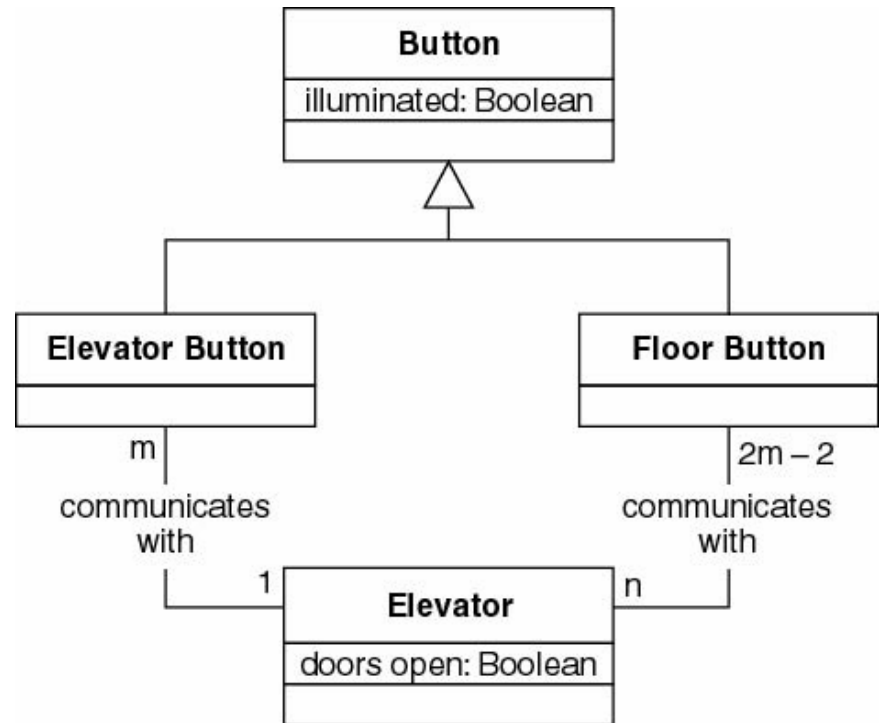
- Nouns outside the problem
 - Floor
 - Building
 - Door

Noun Extraction

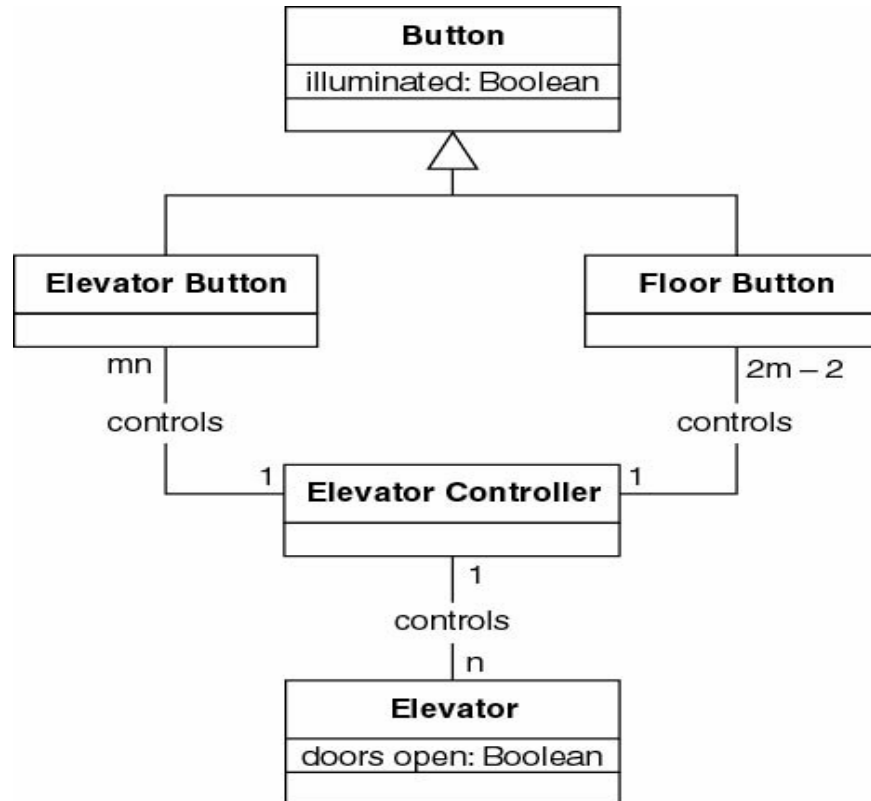
- Abstract nouns
 - Identifies ideas or quantities that have no physical existence
 - Rule of thumb:
 - Abstract nouns rarely end up corresponding to classes
 - Frequently are attributes of classes
 - Movement: attribute of elevator
 - Illumination: attribute of button
 - Request: attribute of user

Noun Extraction

- Remaining nouns
 - Elevator
 - Button



Noun Extraction



CRC Cards

- Class-Responsibility-Collaboration Cards
 - For each class the following are written on a card
 - The name of the class
 - Functionality or responsibility of the class
 - List of classes it invokes or collaborates with

CRC Cards

- Extensions and modifications to CRC Card
 - In addition to responsibility of the class, CRC Card contains attributes and methods of the class
 - Use post-it instead of cards
 - CASE tools

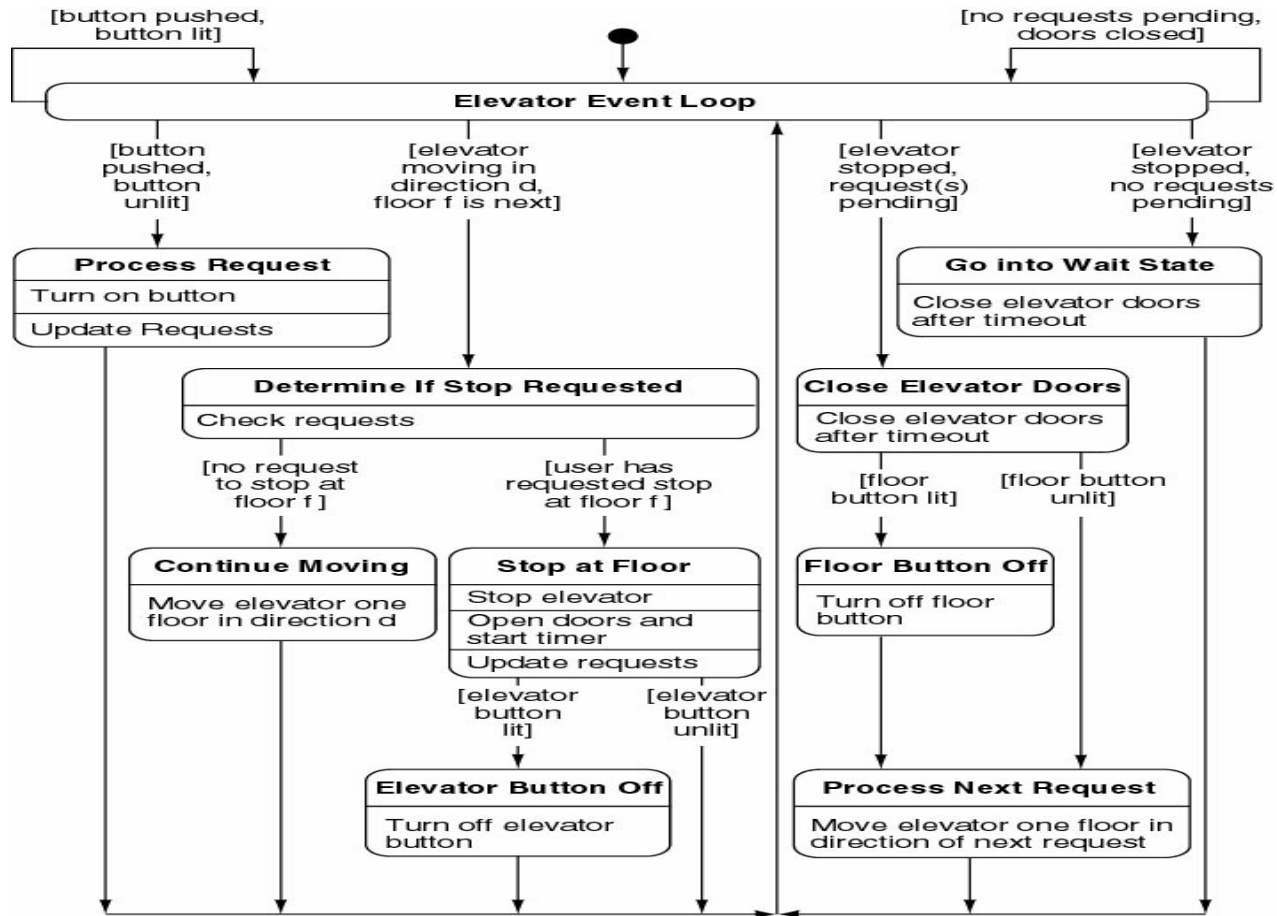
CRC Cards

- Weaknesses
 - Not a good way to identify classes unless the team is very experienced in the application domain

CRC Cards

- Strengths
 - CRC Card can be an excellent tool to insure completeness once the developers have determined the classes and have a good idea of
 - class responsibilities
 - class collaborations
 - Cards can be passed to developers
 - Interaction between teams can uncover missing or incorrect attributes or methods

Dynamic Modeling



Testing During OOA Phase

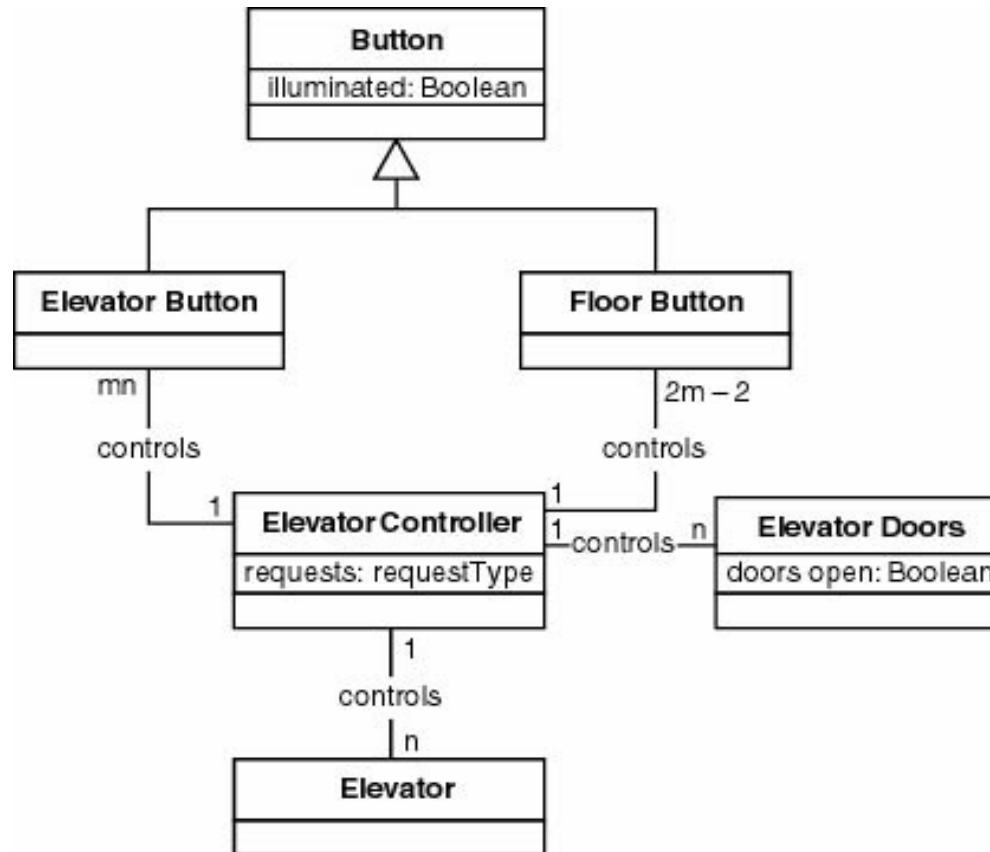
- One component of reviewing the OOA is CRC Card
- Overlooked aspects
 - Responsibility 1
 - Responsibility 7

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

Testing During OOA Phase

CLASS
Elevator Controller
RESPONSIBILITY
<ol style="list-style-type: none">1. Send message to Elevator Button to turn on button2. Send message to Elevator Button to turn off button3. Send message to Floor Button to turn on button4. Send message to Floor Button to turn off button5. Send message to Elevator to move up one floor6. Send message to Elevator to move down one floor7. Send message to Elevator Doors to open8. Start timer9. Send message to Elevator Doors to close after timeout10. Check requests11. Update requests
COLLABORATION
<ol style="list-style-type: none">1. Subclass Elevator Button2. Subclass Floor Button3. Class Elevator Doors4. Class Elevator

Testing During OOA Phase



Testing During OOA Phase

1. User A presses the Up floor button at floor 3 to request an elevator. User A wishes to go to floor 7.
2. The floor button informs the elevator controller that the floor button has been pushed.
3. The elevator controller sends a message to the Up floor button to turn itself on.
4. The elevator controller sends a series of messages to the elevator to move itself up to floor 3. The elevator contains User B, who has entered the elevator at floor 1 and pressed the elevator button for floor 9.
5. The elevator controller sends a message to the Up floor button to turn itself off.
6. The elevator controller sends a message to the elevator doors to open themselves.
7. The elevator control starts the timer.
User A enters the elevator.
8. User A presses elevator button for floor 7.
9. The elevator button informs the elevator controller that the elevator button has been pushed.
10. The elevator controller sends a message to the elevator button for floor 7 to turn itself on.
11. The elevator controller sends a message to the elevator doors to close themselves after a timeout.
12. The elevator controller sends a series of messages to the elevator to move itself up to floor 7.
13. The elevator controller sends a message to the elevator button for floor 7 to turn itself off.
14. The elevator controller sends a message to the elevator doors to open themselves to allow User A to exit from the elevator.
15. The elevator controller starts the timer.
User A exits from the elevator.
16. The elevator controller sends a message to the elevator doors to close themselves after a timeout.
17. The elevator controller sends a series of messages to the elevator to move itself up to floor 9 with User B.

CASE Tools for the OOA Phase

- CASE support the graphical aspects of OOA
- Strengths
 - Change to the model is reflected automatically in affected diagrams
 - CASE tools support other parts of the OO life cycle
- Examples
 - Rose
 - Together

Challenges of the OOA Phase

- Document must be simple and complete
- Transition between OOA and OOD is smoother than the transition in the classical paradigm
 - Easy to cross boundaries between analysis (WHAT) and design (HOW)
- Essential to remember that the OOA is an iterative process

Challenges of the OOA Phase

- Initial part of classical design phase is to decompose the product
- On the contrary, “classes” and the “modules” of the OOD phase were extracted during OOA phase
- Temptation of carrying into design phase is extremely high since the classes are present from early on