

Chapter 6 (Part 2):

Requirements Modeling: Class-Based Modeling

Shifting from external user views (Use Cases)
to internal conceptual objects.

Introduction & Purpose



The Core Question:

What are the fundamental "things" inside the system that have both data and behavior?

Name
Attributes
Operations

Analysis Class:

A conceptual representation of a system element encapsulating data (attributes) and behavior (operations).



The Bridge:

These classes form the core of the Analysis Model and directly evolve into Design Classes.



Goal:

To define the object-oriented structure of the system at a conceptual level.

Learning Objectives

By the end of this lecture, you will be able to:

- Identify potential analysis classes using grammatical parsing and behavioral analysis.
- Specify attributes and operations for these classes.
- Create CRC (Class-Responsibility-Collaborator) cards for exploratory modeling.
- Define associations, dependencies, and analysis packages.

Identifying Analysis Classes: Method 1 (Grammatical Parse)

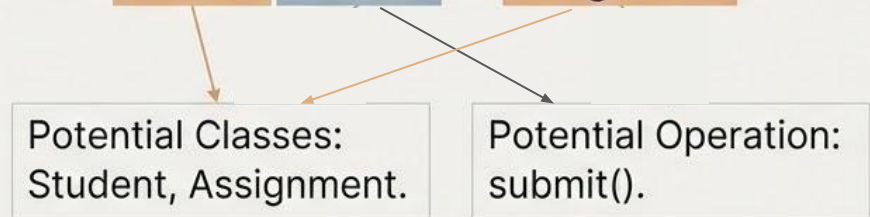
Process: Examine the text of use cases or requirements statements.

The Rules:

- **Nouns/Noun Phrases:** Indicate potential Analysis Classes or Attributes.
- **Verbs:** Indicate potential Operations or Relationships.

Example:

“The student submits an assignment...”



The SafeHome security function enables the homeowner to configure the security system when it is installed, monitors all sensors connected to the security system, and interacts with the homeowner through the Internet, a PC, or a control panel.

During installation, the SafeHome PC is used to program and configure the system. Each sensor is assigned a number and type, a master password is programmed for arming and disarming the system, and telephone number(s) are input for dialing when a sensor event occurs.

When a sensor event is recognized, the software invokes an audible alarm attached to the system. After a delay time that is specified by the homeowner during system configuration activities, the software dials a telephone number of a monitoring service, provides information about the location, reporting the nature of the event that has been detected. The telephone number will be redialed every 20 seconds until telephone connection is obtained.

The homeowner receives security information via a control panel, the PC, or a browser, collectively called an interface. The interface displays prompting messages and system status information on the control panel, the PC, or the browser window. Homeowner interaction takes the following form . . .

Extracting the nouns, we can propose a number of potential classes:

Potential Class	General Classification
homeowner	role or external entity
sensor	external entity
control panel	external entity
installation	occurrence
system (alias security system)	thing
number, type	not objects, attributes of sensor
master password	thing
telephone number	thing
sensor event	occurrence
audible alarm	external entity
monitoring service	organizational unit or external entity

Identifying Analysis Classes: Method 2 (CRC Modeling)

CRC: Class-Responsibility-Collaborator Modeling.

Format:

- A collaborative, exploratory technique using index cards.

Components:

- **Responsibilities:** What the class knows or does.
- **Collaborators:** Other classes it works with to fulfill responsibilities.

CRC Card	
Responsibilities What it knows/does	Collaborators Other classes it works with

Process:

- Brainstorm classes, create cards, and role-play scenarios.

CRC Card Example

Class Name: Student

CRC Card	
Responsibilities Submit assignment View grades	Collaborators Assignment CourseGrade

Note: Collaborators listed here directly indicate relationships (associations/dependencies).

Defining Class Internals: Attributes

Attributes: Data held by the object.

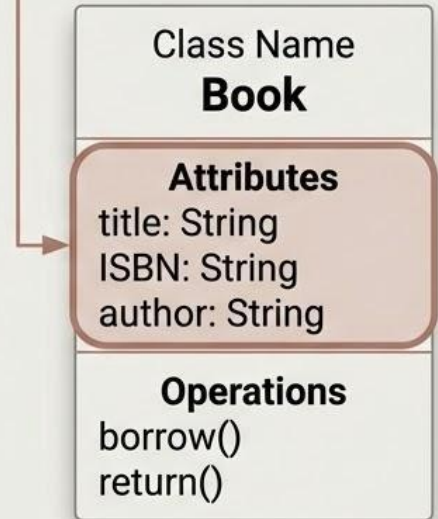
Definition:

- A named property describing a data value held by the object.

Guidelines:

- Ask: What data does this class need to remember?
- Focus on essential domain attributes (e.g., Book: ISBN, Title).
- Avoid: Implementation details (e.g., pointers, database IDs).

Notation: Listed in the middle compartment of a UML class box.



Note: Attributes represent data, not behavior.

Defining Class Internals: Operations

Operations: Object
behavior & services.

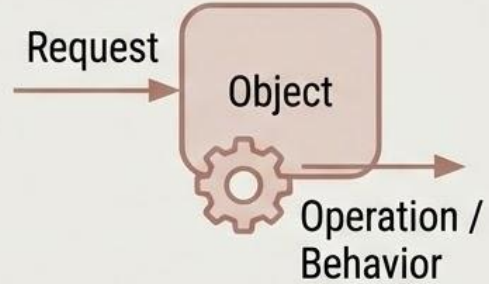
Definition:

A service that can be requested from an object; implies behavior.

Sources for Operations:

- Use Case “System Response” steps.
- CRC Card Responsibilities.
- Class Lifecycle (creation, deletion) and “Housekeeping.”

Guidelines: Use strong verbs (e.g., calculateTotal(), validate()) and avoid low-level get/set details.



UML Class Notation Example

Visualizing a 'Book' Class

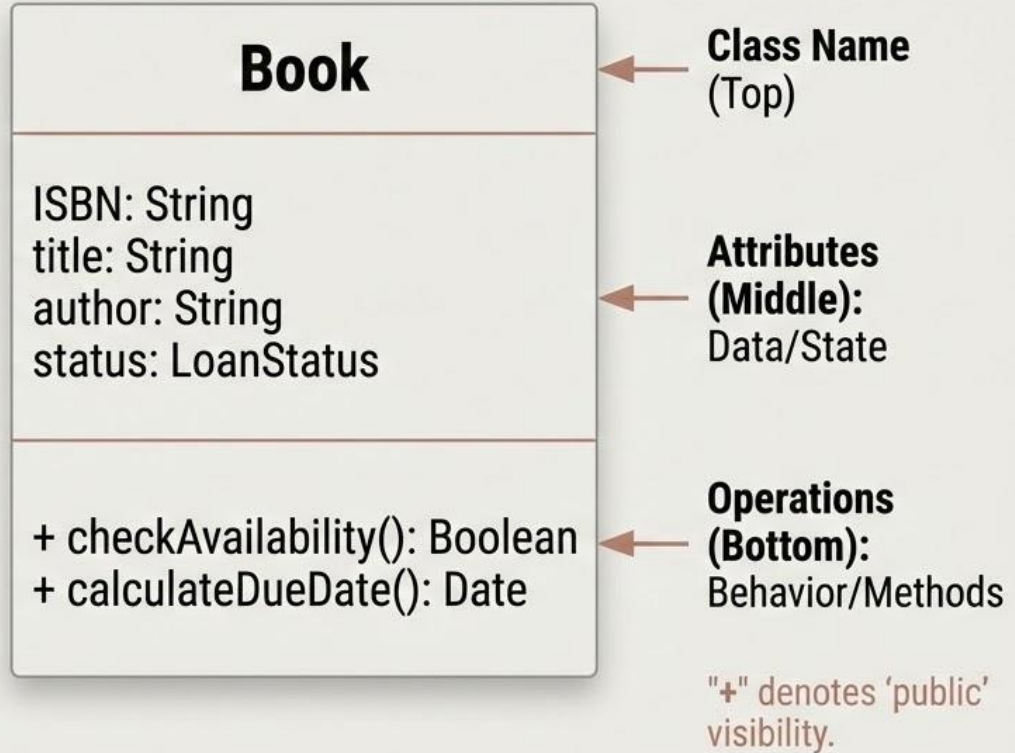


FIGURE 6.9

Class diagram
for the system
class

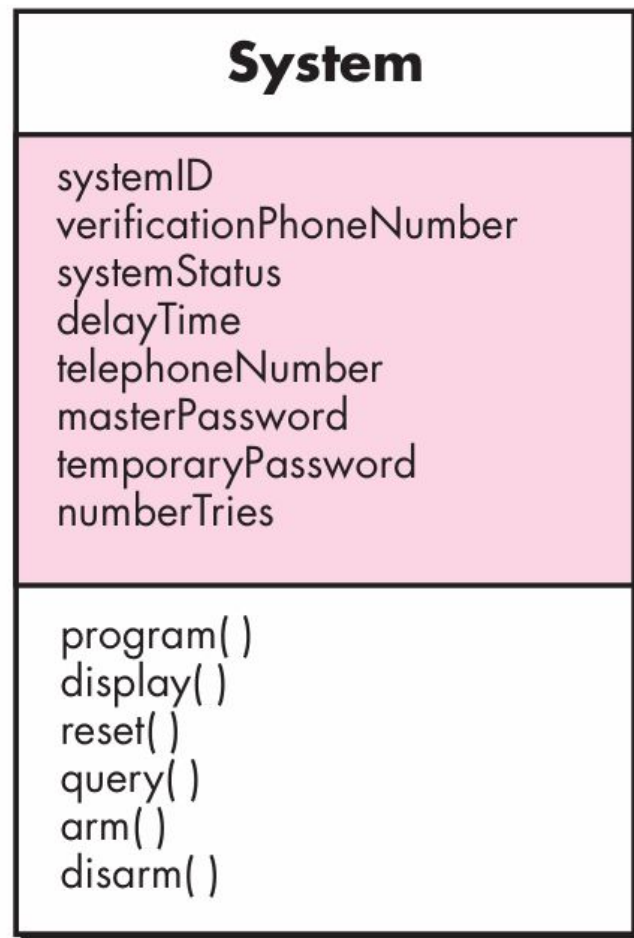


FIGURE 6.10

Class diagram
for FloorPlan
(see sidebar
discussion)

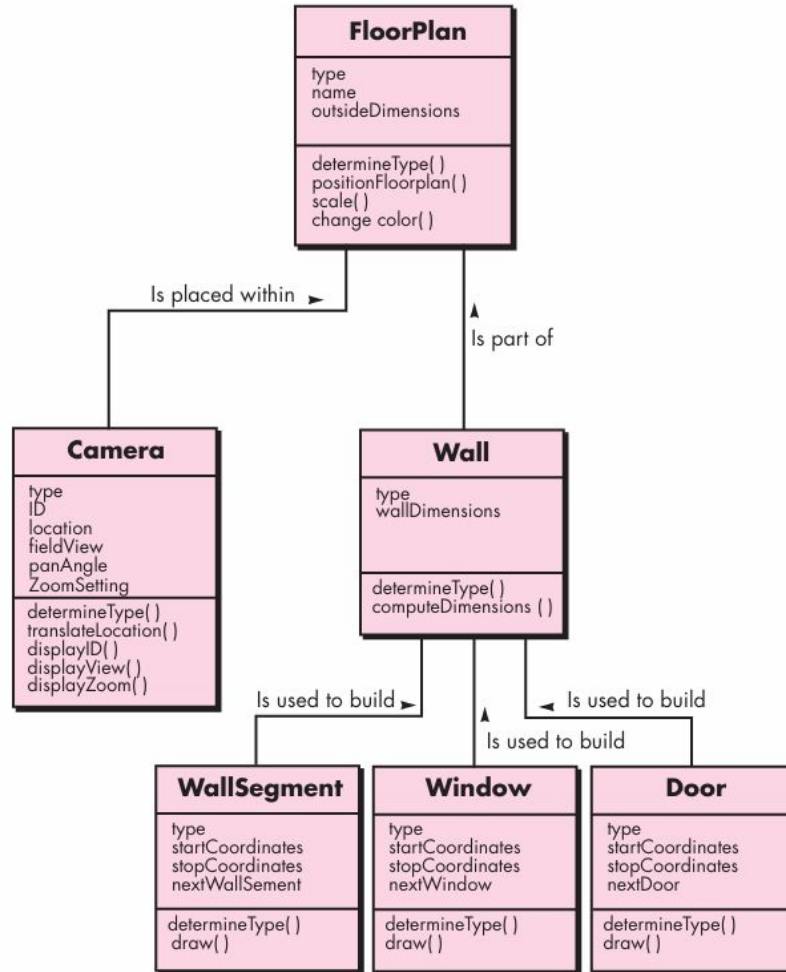


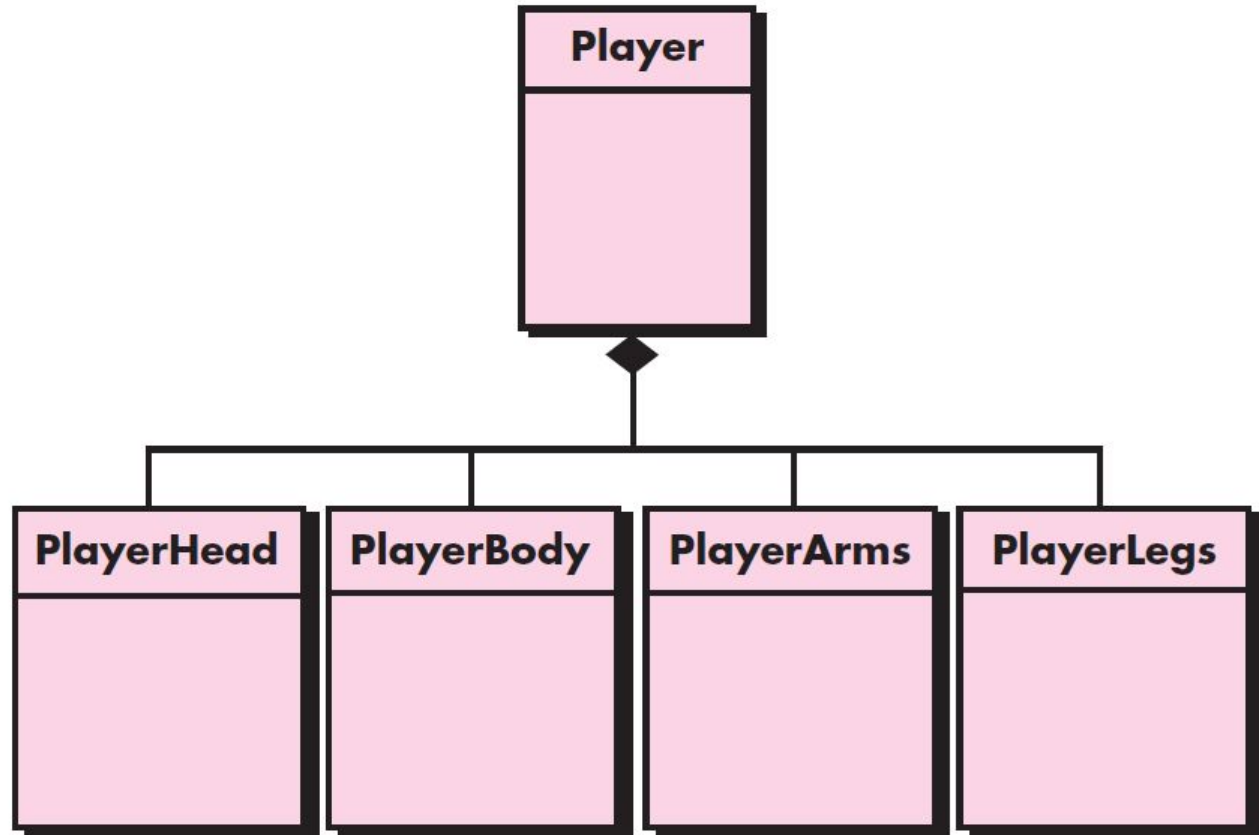
FIGURE 6.11

A CRC model
index card

Class: FloorPlan	
Description	
Responsibility:	Collaborator:
Defines floor plan name/type	
Manages floor plan positioning	
Scales floor plan for display	
Scales floor plan for display	
Incorporates walls, doors, and windows	Wall
Shows position of video cameras	Camera

FIGURE 6.12

A composite
aggregate
class



Class Relationships: Associations

A structural relationship indicating a meaningful, persistent connection.

Key Feature: Multiplicity (Cardinality).

- 1 (Exactly one)
- * or 0..* (Many)
- 1..* (One or more)

Example & Visual:



Example: A Library has Books (Library 1 --- * Book).

Visual: Solid line connecting classes.

Class Relationships: Dependencies

A "using" relationship where a change to the supplier affects the client.

Characteristics:

- Weaker and more transient than an association.
- Client uses the supplier but does not persistently maintain a link.

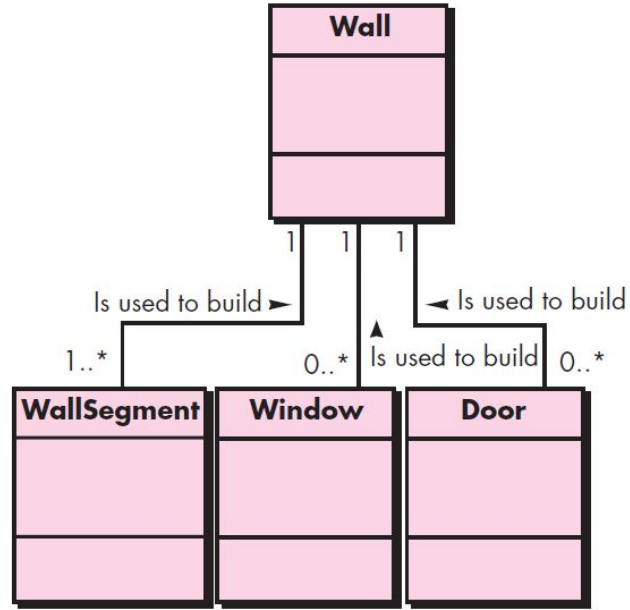
Example & Visual:



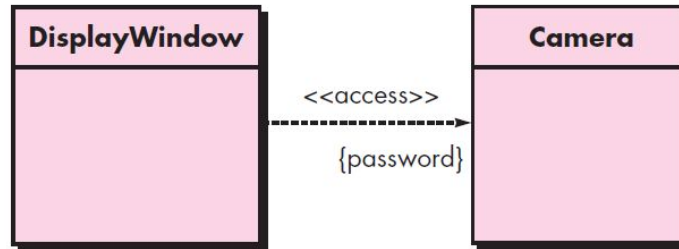
Example: **ReportGenerator** depends on **DataFormatter** to do work but doesn't store it.

FIGURE 6.13

Multiplicity

**FIGURE 6.14**

Dependencies



Managing Complexity: Analysis Packages

Problem: As the number of classes grows, complexity increases.

Solution: Analysis Package—a grouping of logically related classes.

Grouping Criteria:

- Classes that collaborate heavily (high coupling).
- Classes that support a specific business function (e.g., LoanManagement).

Visual Representation:



Visual: A tabbed folder icon grouping related classes.

Summary & Synthesis

The Workflow:

- **Parse Use Cases:** Nouns → Classes; Verbs → Operations.
- **CRC Modeling:** Discover responsibilities and collaborators.
- **Refine:** Define Attributes and Relationships (Associations/Dependencies).
- **Organize:** Group into Analysis Packages.

“

Key Takeaway:

A good class model captures the "soul" of the problem domain and creates a shared vocabulary.

”

Assignment: Hotel Reservation System



Task: Apply today's concepts to a Hotel System.

- **CRC:** Create a card for the Reservation class (2 responsibilities, 2 collaborators).
- **Attributes/Operations:** Specify 3 attributes and 1 key operation for the Room class.
- **Relationships:** Describe the association (with multiplicity) between Guest and Reservation.