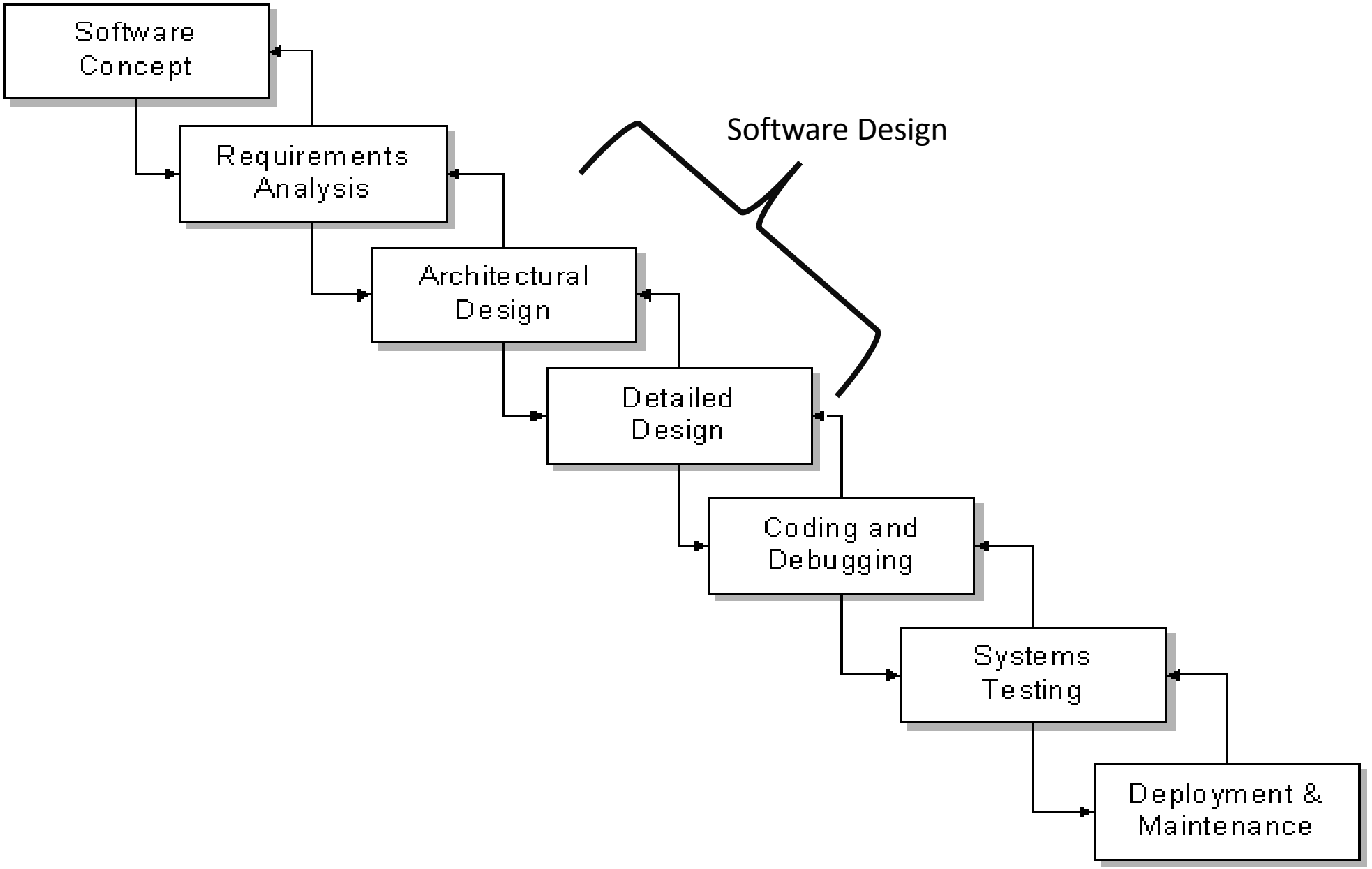
University of
Asia Pacific

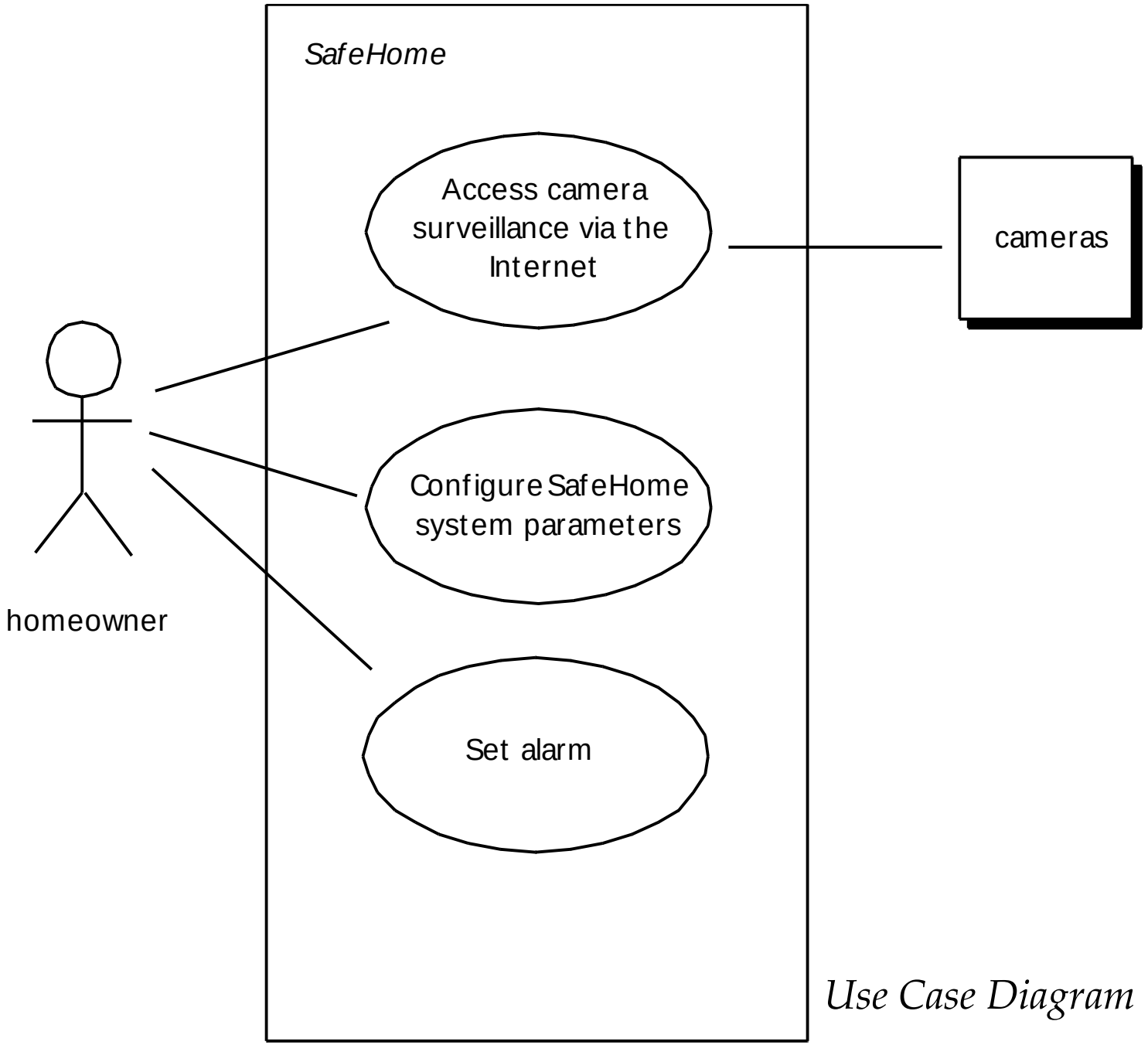
CSE 321Software Engineering

Software Design

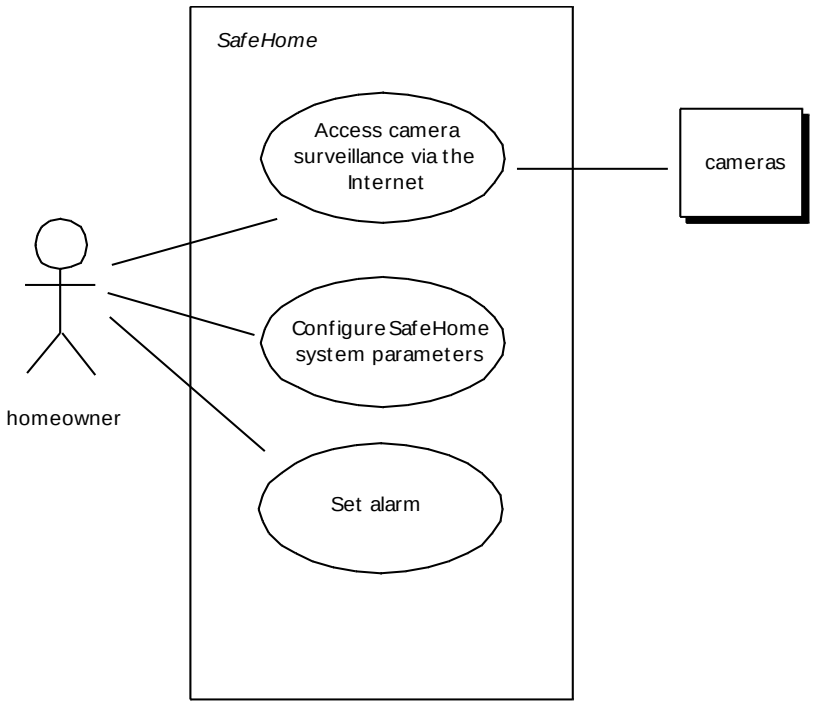
Md. Shamsul Haque

Product(Software) Engineering(development) Process

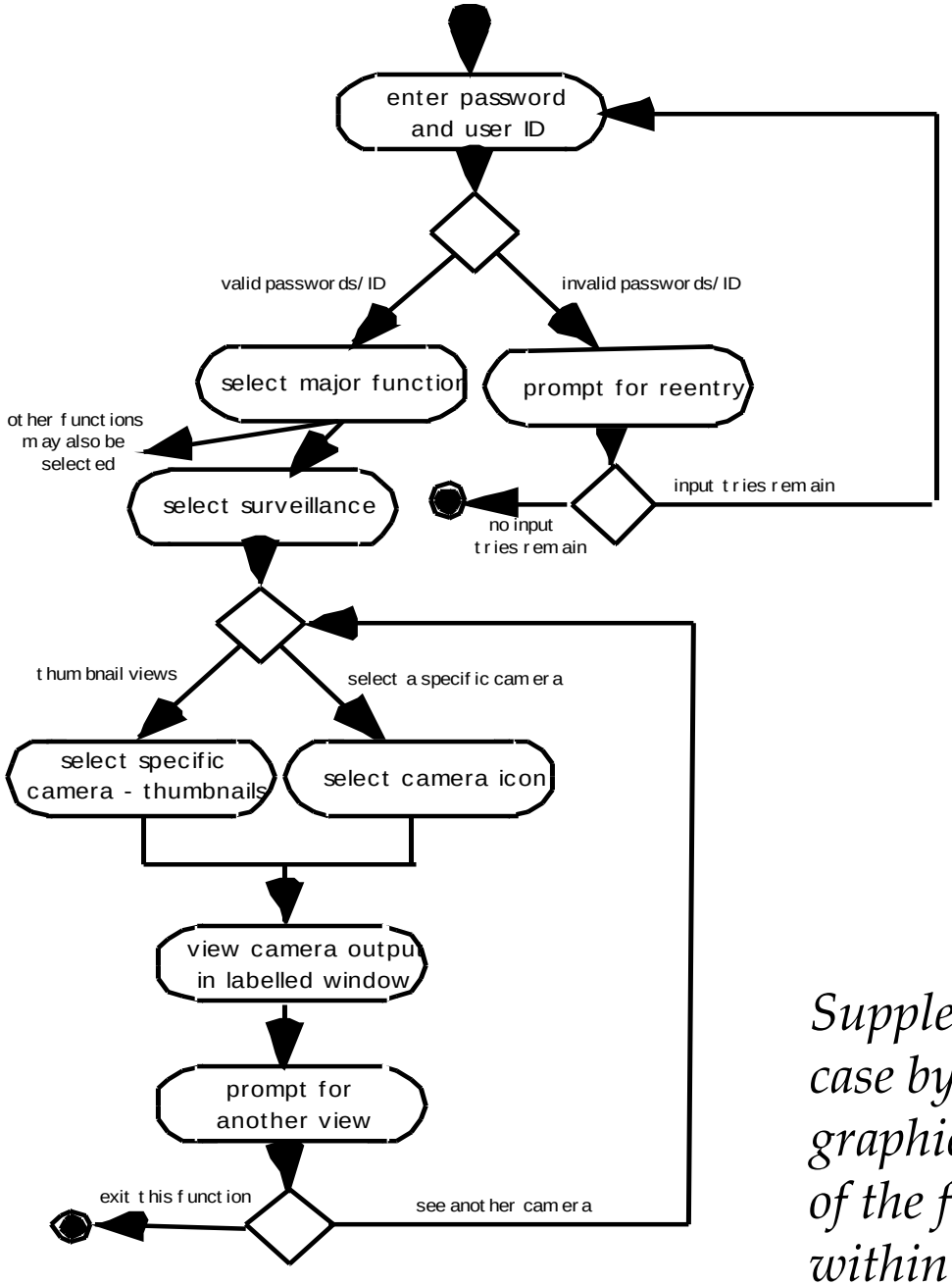




Use Case Diagram

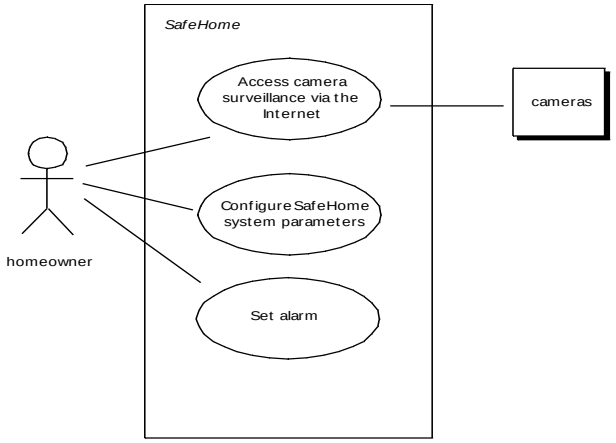


Use Case Diagram

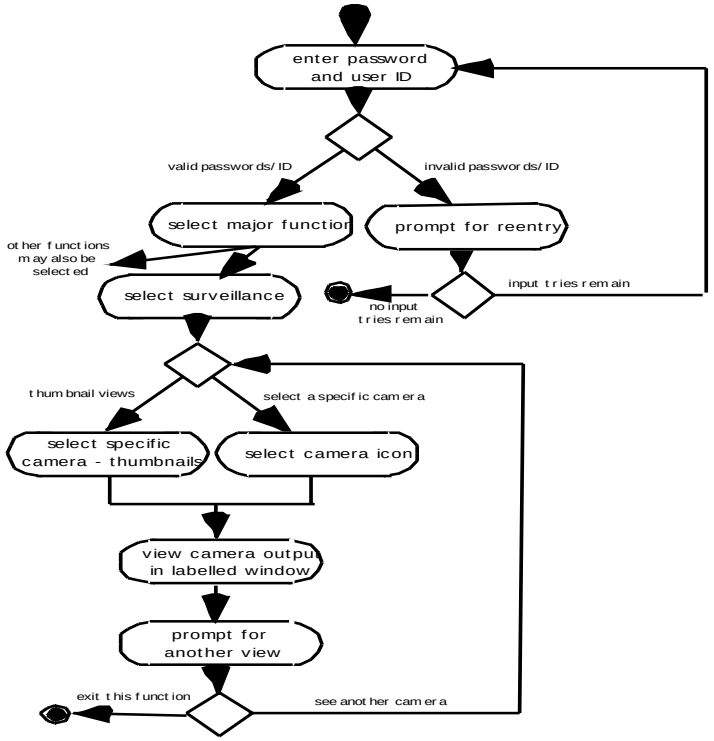


Activity Diagram

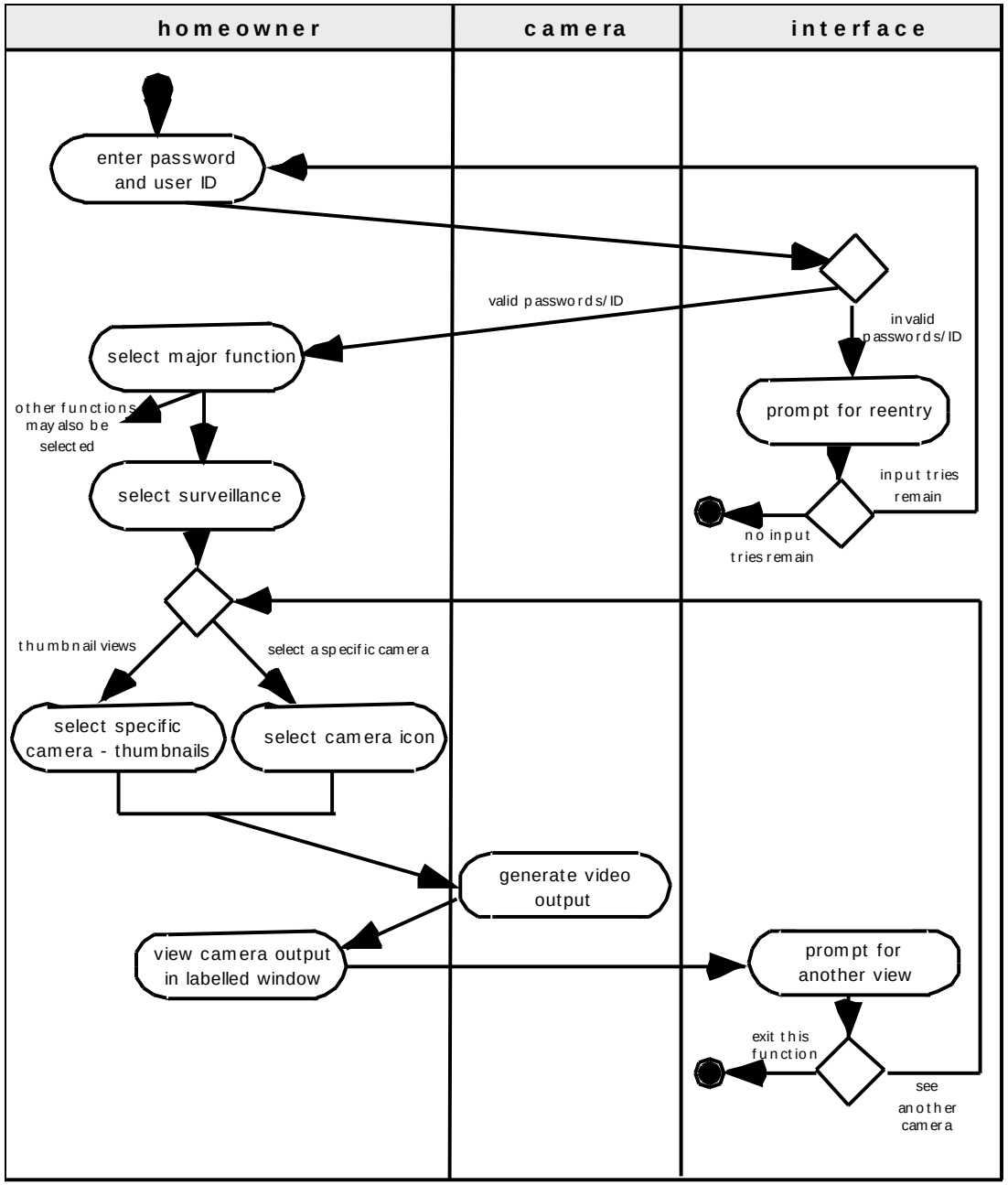
Supplements the use case by providing a graphical representation of the flow of interaction within a specific scenario



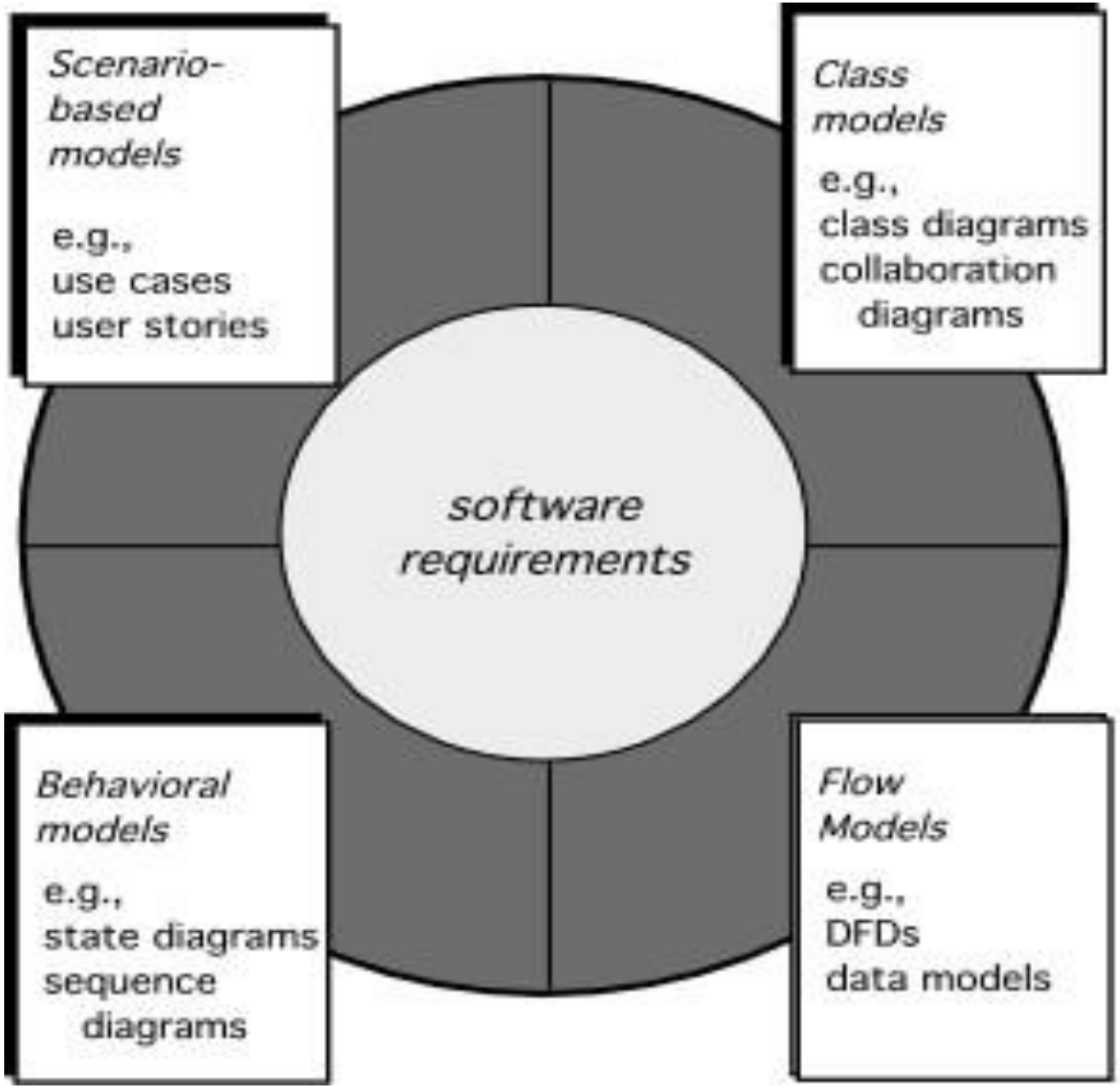
Use Case Diagram



Activity Diagram



Swimlane Diagram



Elements of Analysis Modeling

Data Modeling / Class Modeling

- examines data objects independently of processing
- focuses attention on the data domain
- creates a model at the customer’s level of abstraction
- indicates how data objects relate to one another

Data Objects and Attributes

A data object contains a set of attributes that act as an aspect, quality, characteristic, or descriptor of the object

object: automobile

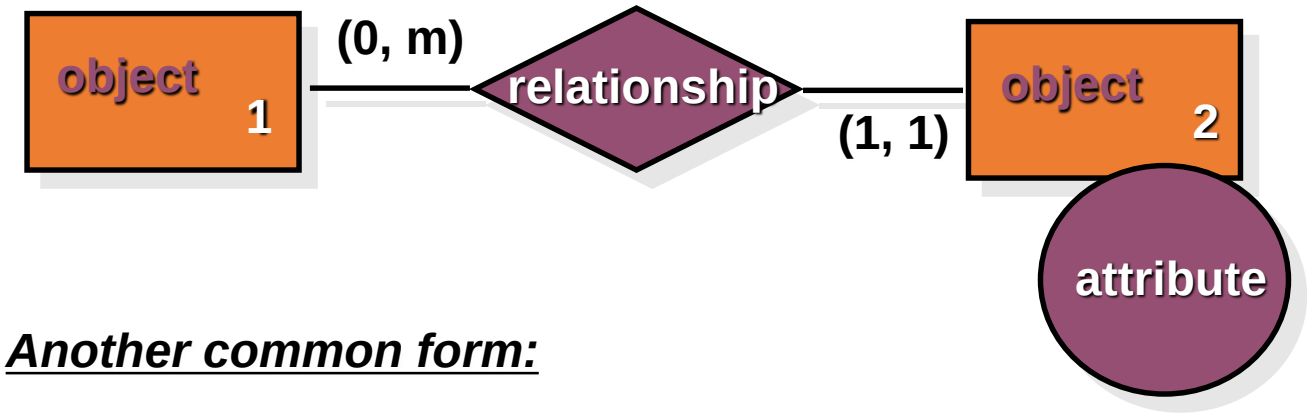
attributes:
- **make**
- **model**
- **body type**
- **price**
- **options code**

What is a Relationship?

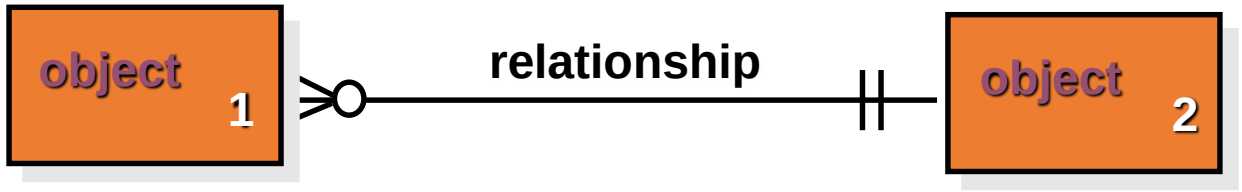
- Data objects are connected to one another in different ways.
  - A connection is established between **person** and **car** because the two objects are related.
    - A person *owns* a car
    - A person *is insured to drive* a car
- The relationships *owns* and *insured to drive* define the relevant connections between **person** and **car.**
- Several instances of a relationship can exist
- Objects can be related in many different ways

ERD Notation

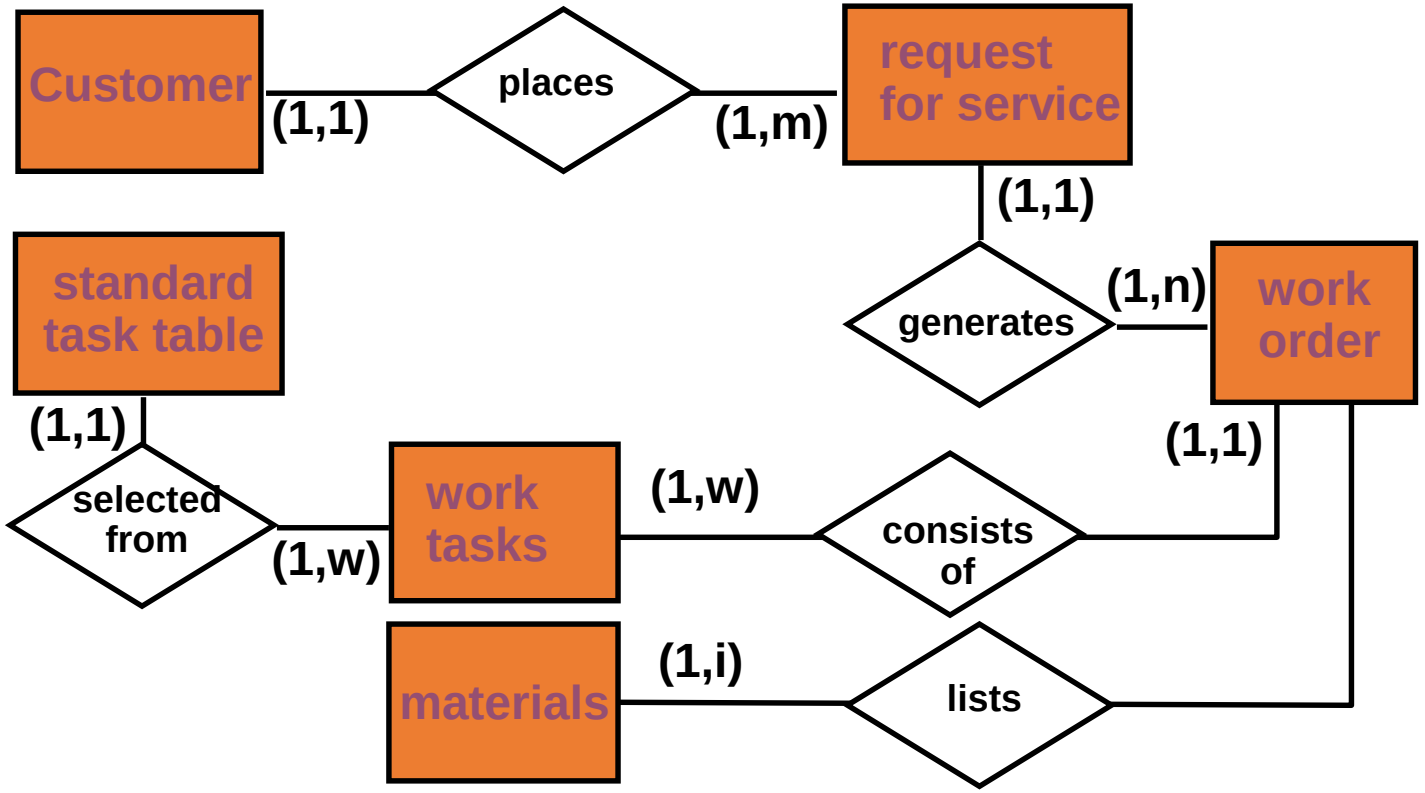
One common form:



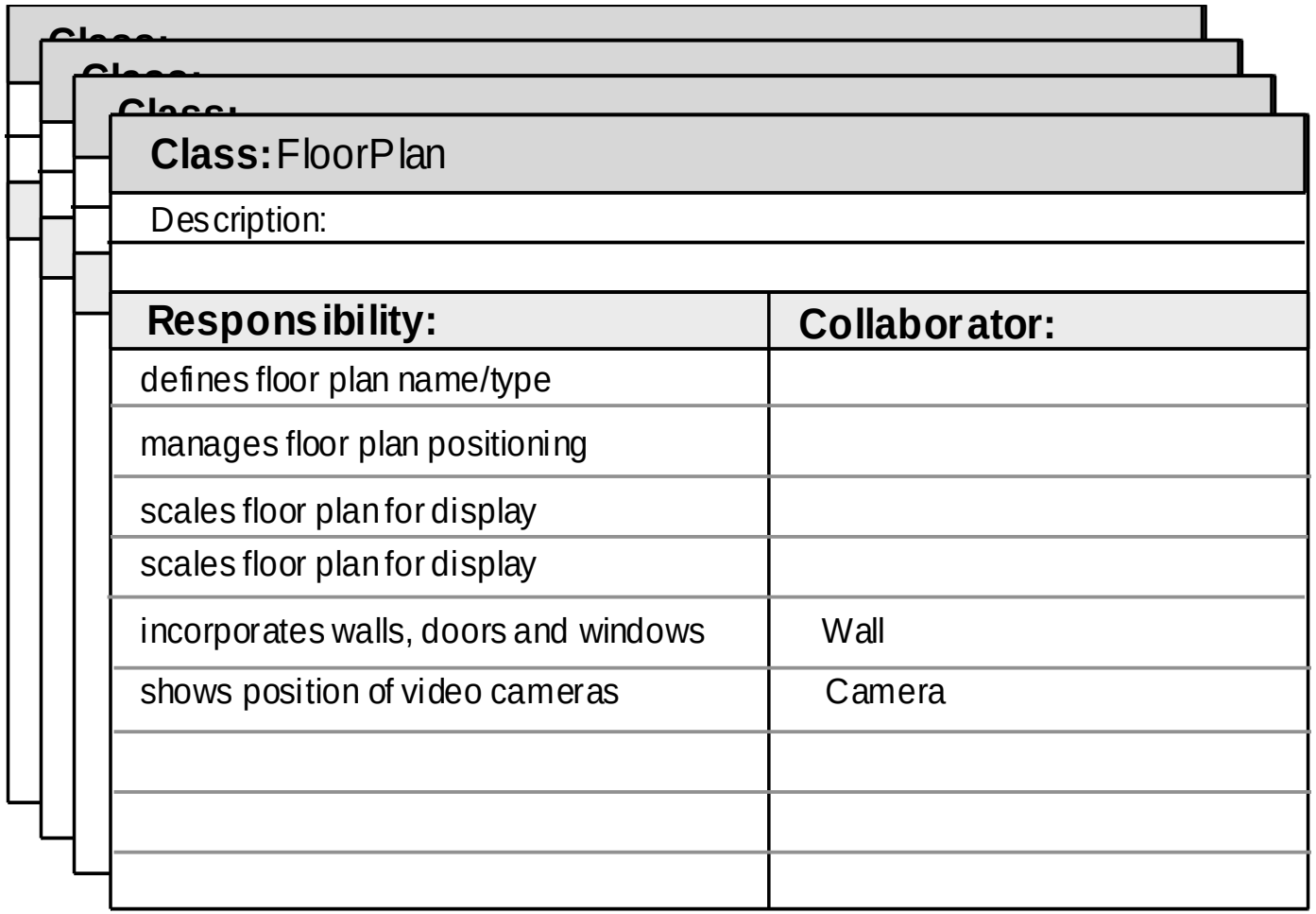
Another common form:



The ERD: An Example



CRC Modeling



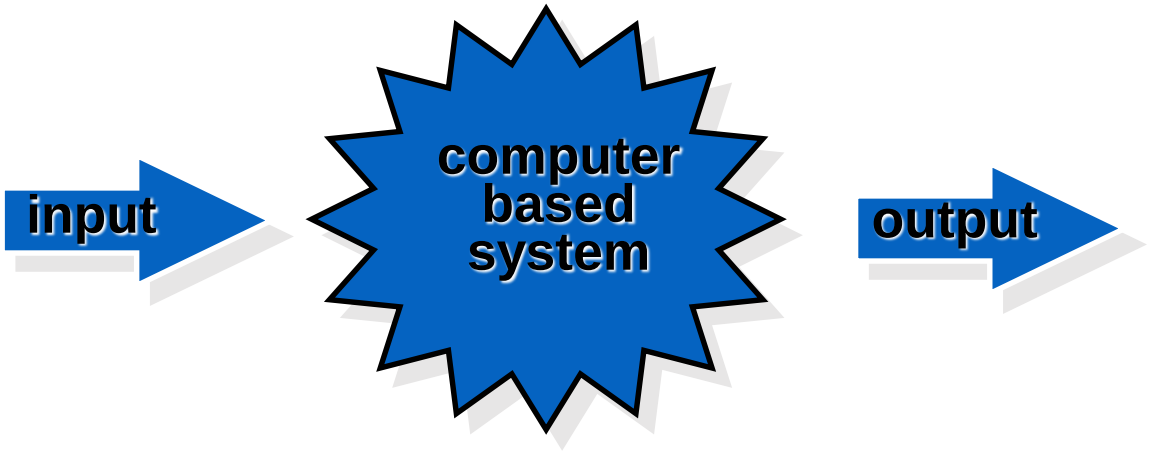
Class: FloorPlan	
Description:	
Responsibility:	**Collaborator:**
defines floor plan name/type	
manages floor plan positioning	
scales floor plan for display	
scales floor plan for display	
incorporates walls, doors and windows	Wall
shows position of video cameras	Camera

Flow-Oriented Modeling

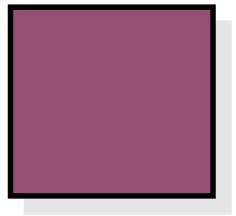
- Represents how data objects are transformed at they move through the system
- **data flow diagram (DFD)** is the diagrammatic form that is used
- Considered by many to be an "old school" approach, but continues to provide a view of the system that is unique—it should be used to supplement other analysis model elements

The Flow Model

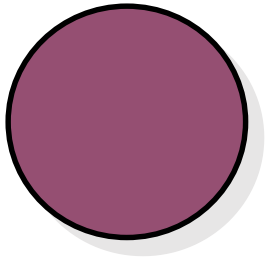
Every computer-based system is an information transform ....



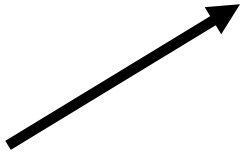
Flow Modeling Notation



external entity *Examples:* a person, a device, a sensor



process Examples: *compute taxes, determine area, format report, display graph*



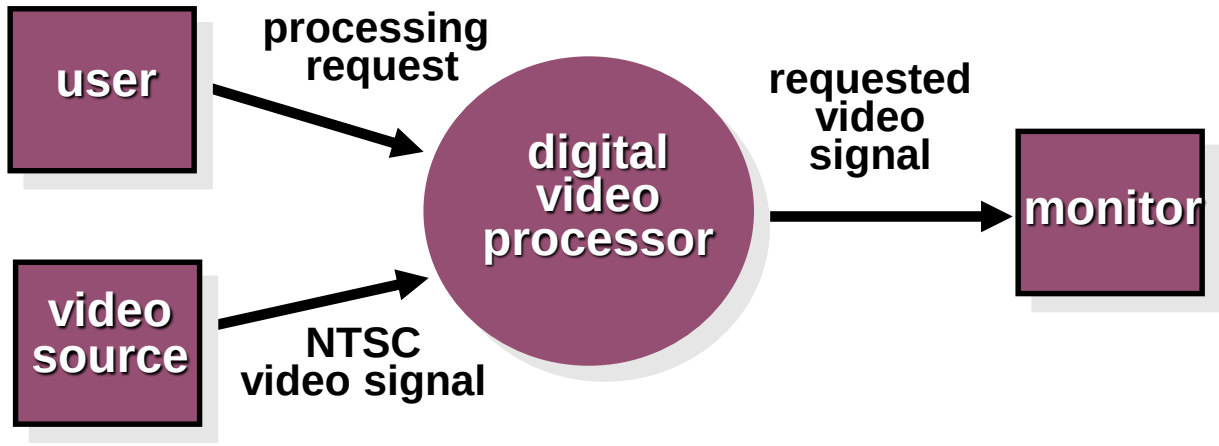
data flow Flow direction

data store Storage

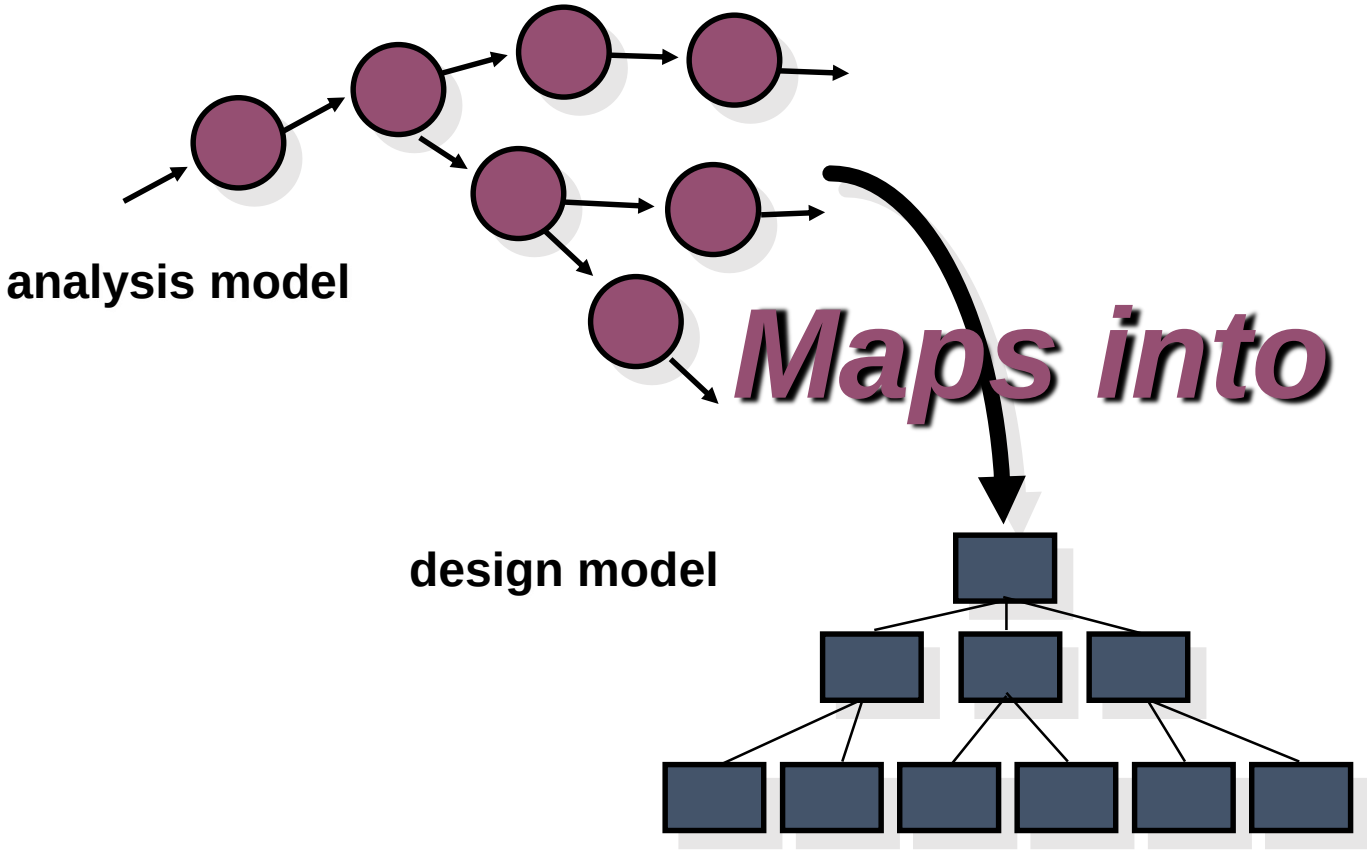
Data Flow Diagramming: Guidelines

- all icons must be labeled with meaningful names
- the DFD evolves through a number of levels of detail
- always begin with a context level diagram (also called level 0)
- always show external entities at level 0
- always label data flow arrows
- do not represent procedural logic

Level 0 DFD Example



DFDs: A Look Ahead



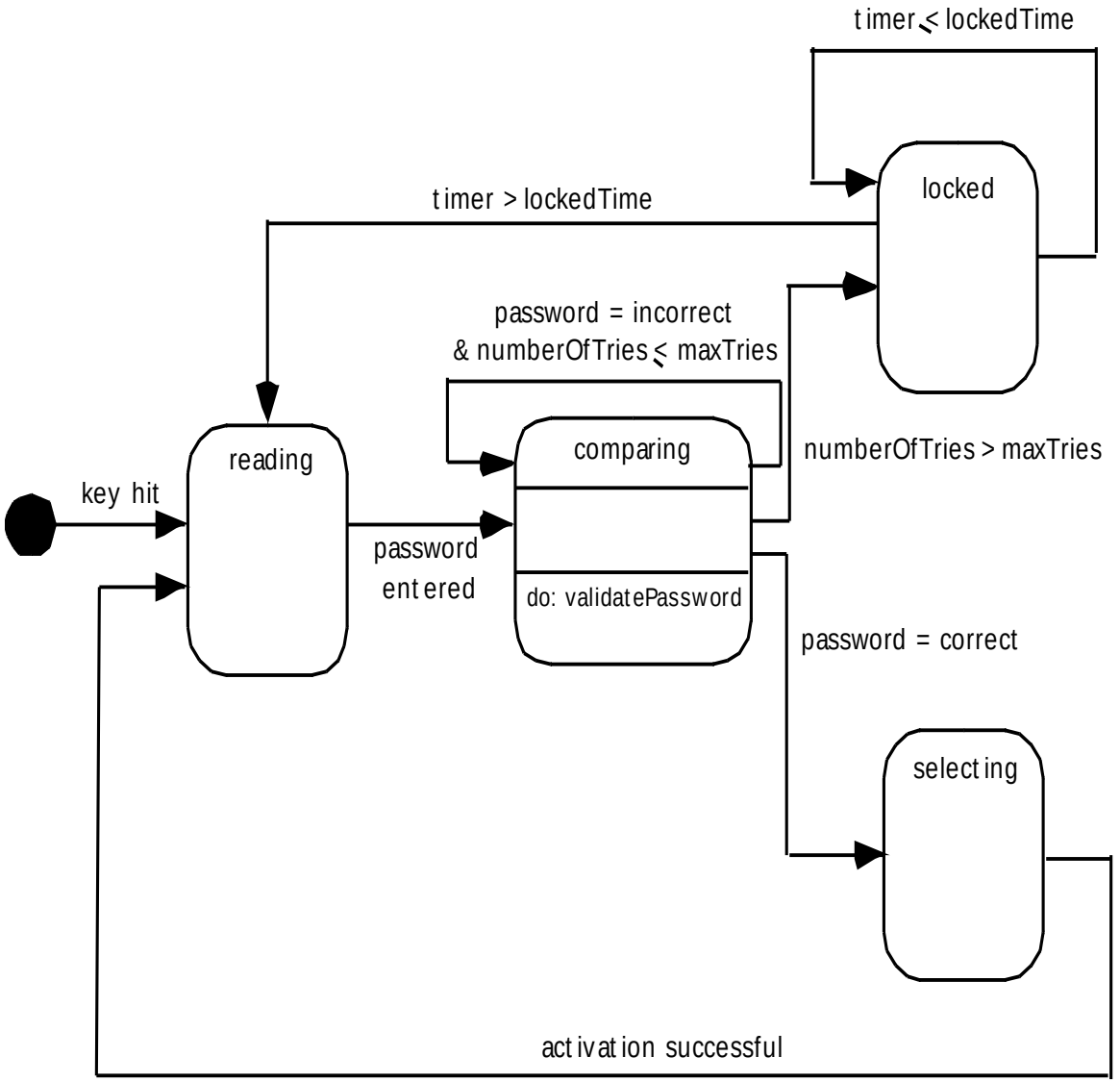
Behavioral Modeling

- The behavioral model indicates how software will respond to external events or stimuli. To create the model, the analyst must perform the following steps:
    - Evaluate all use-cases to fully understand the sequence of interaction within the system.
    - Identify events that drive the interaction sequence and understand how these events relate to specific objects.
    - Create a sequence for each use-case.
    - Build a state diagram for the system.
    - Review the behavioral model to verify accuracy and consistency.

State Representations

- In the context of behavioral modeling, two different characterizations of states must be considered:
  - the state of each class as the system performs its function and
  - the state of the system as observed from the outside as the system performs its function
- The state of a class takes on both passive and active characteristics [CHA93].
  - A *passive state* is simply the current status of all of an object's attributes.
  - The *active state* of an object indicates the current status of the object as it undergoes a continuing transformation or processing.

State Diagram for the ControlPanel Class

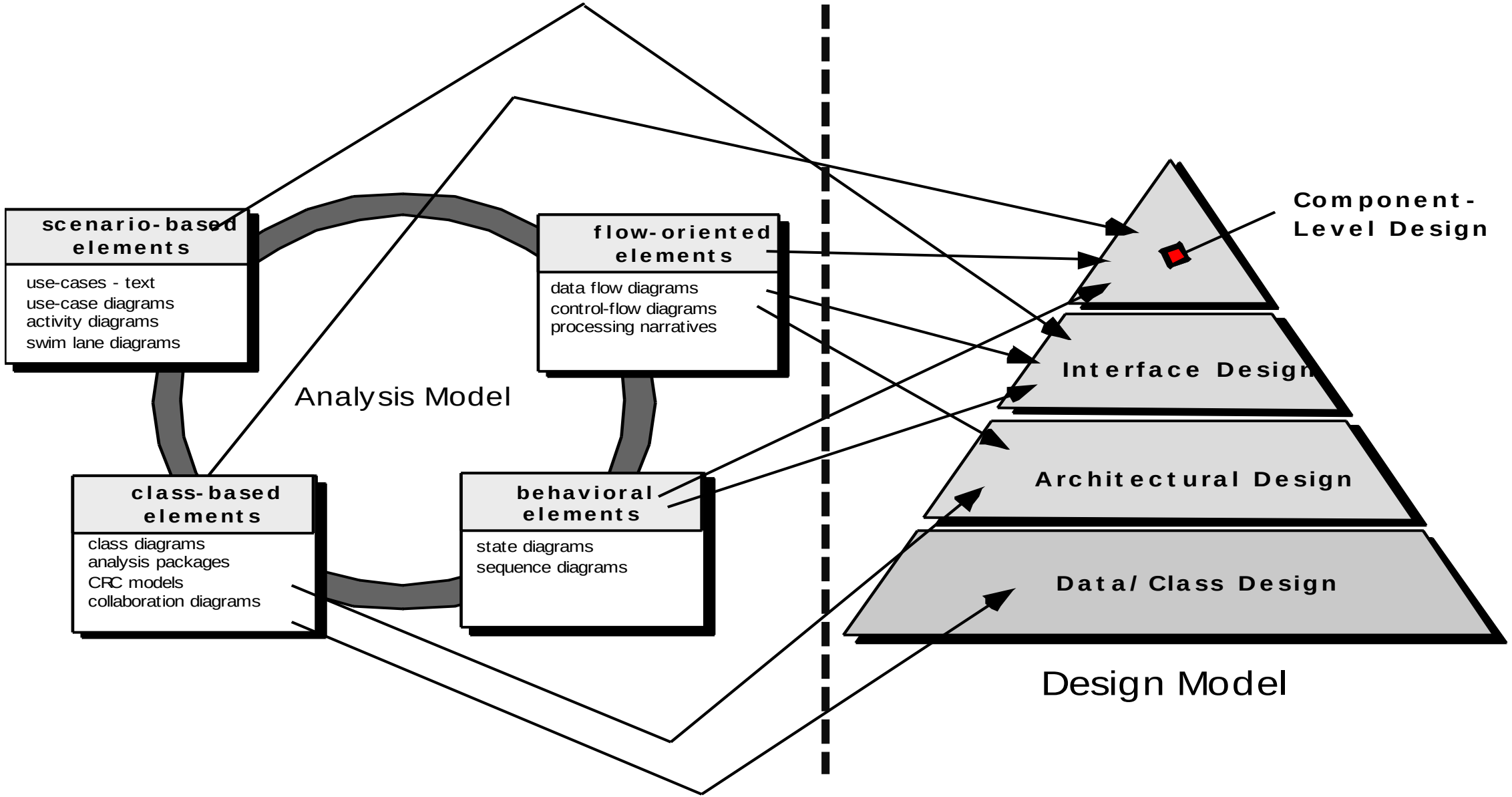


Software Design

- Software design encompasses the set of principles, concepts, and practices that lead to the development of a high-quality system or product.

- Software design sits at the technical kernel of software engineering and is applied regardless of the software process model that is used. Beginning once software re-quirements have been analyzed and modeled, software design is the last software engineering action within the modeling activity and sets the stage for construction (code generation and testing)

Req. Modeling and Software Design



Question
&
Answer Session