분석 모델링

Flow-oriented Modeling

Flow-oriented Models

- Date flow diagrams
- Control-flow diagrams
- Processing narratives

The Flow Model

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

input → **Transformation** → output
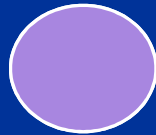
computer-based system

Flow-oriented Model

- The <u>most widely used</u> analysis model today

- Data Flow Diagram (DFD)
  - Data object: Labeled arrows
  - Transformations: Circles (bubbles)
  - DFD is represented in a hierarchical fashion
  - Level 0 DFD or context diagram represents the system as a whole

Flow Modeling Notation

external entity

process

data flow

data store

External Entity

A producer or consumer of data

Examples: a person, a device, a sensor

Another example: computer-based system

Data must always originate somewhere and must always be sent to something
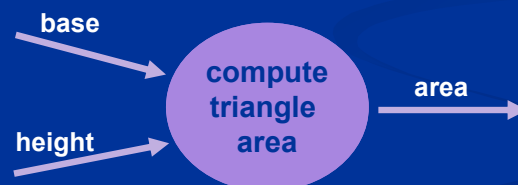
Process

A data transformer (changes input to output)

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

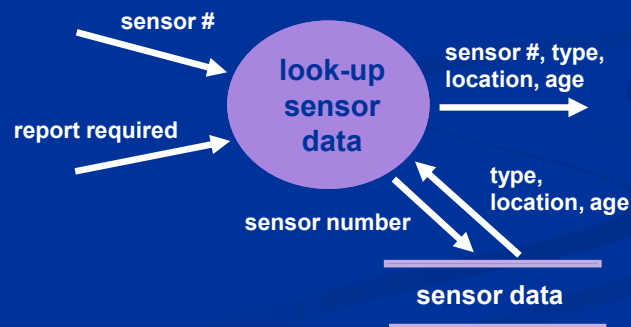*Data must always be processed in some way to achieve system function*

Data Flow

Data flows through a system, beginning as input and be transformed into output.

base

compute triangle area

area

height

Data Stores

**Data is often stored for later use.**
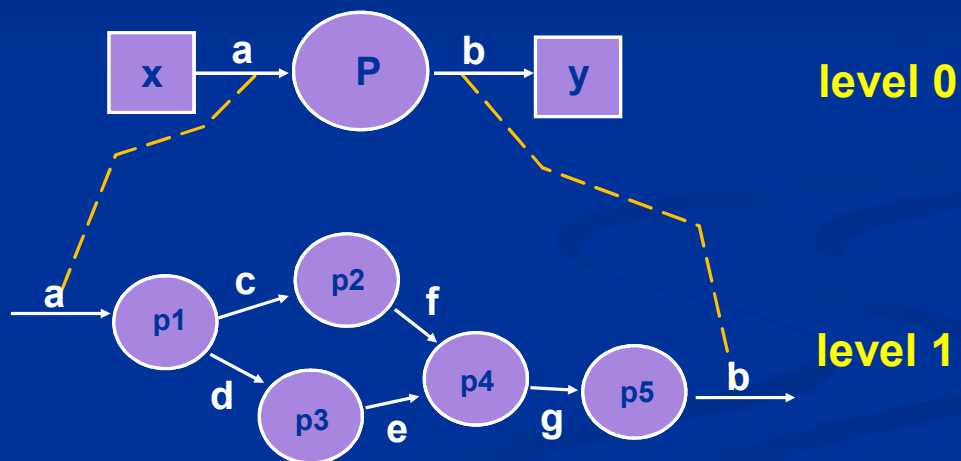

Flow-oriented Model

- Guidelines

1. The level 0 data flow diagram (context diagram) should depict the software/system as a single bubble
2. Primary input and output should be carefully noted
3. Refinement should begin by isolating candidate processes, data objects, and data stores to be represented at the next level
4. All arrows and bubbles should be labeled with meaningful names
5. Information flow continuity must be maintained from level to level
6. One bubble at a time should be refined
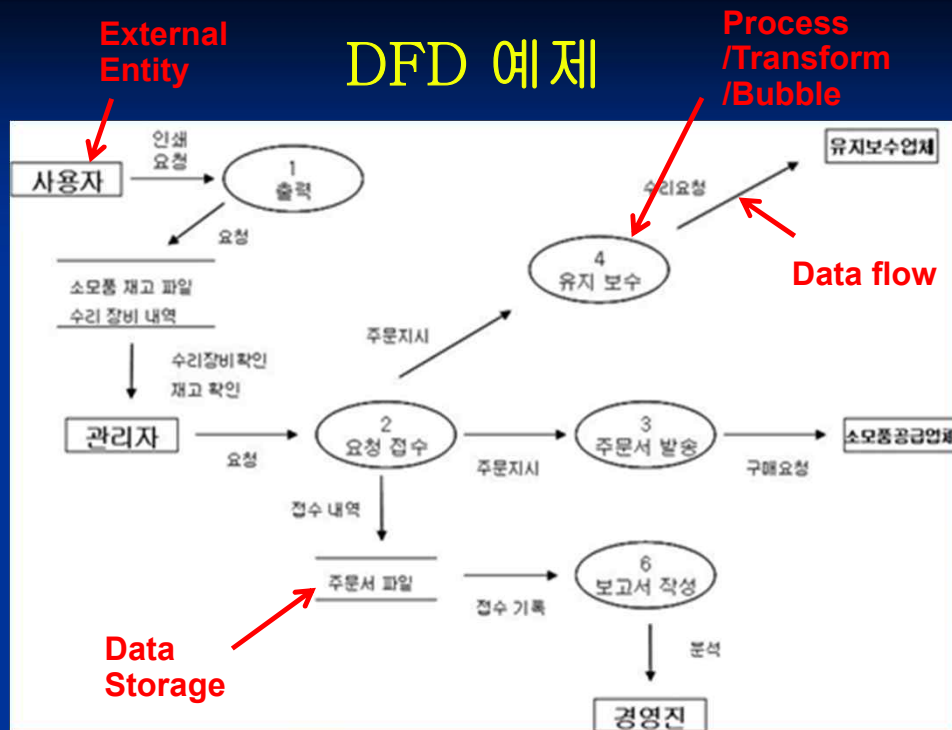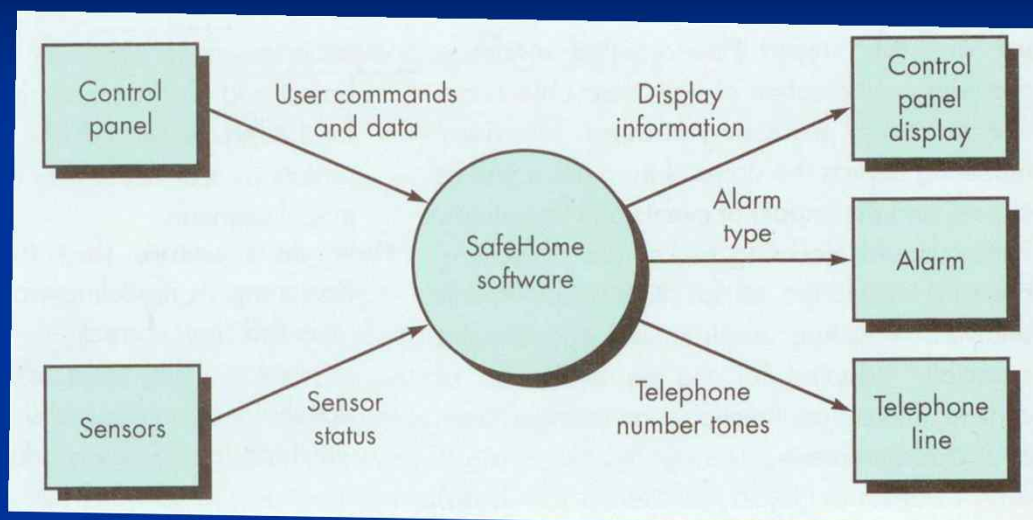
Level 0 DFD
(배경도: Context Diagram)
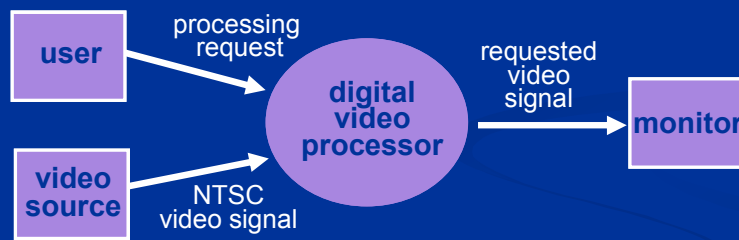
The Data Flow Hierarchy

DFD 예제

External Entity

Process /Transform /Bubble
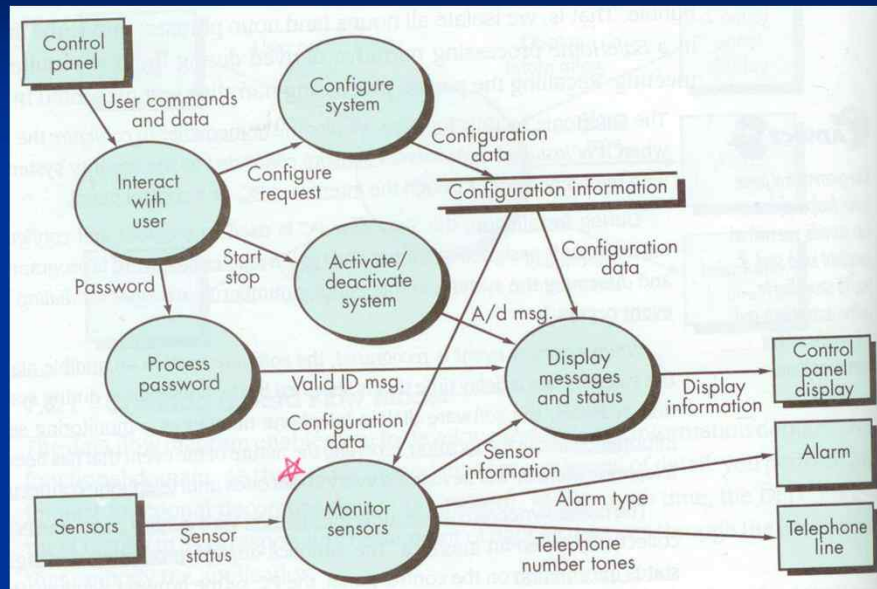
Data flow
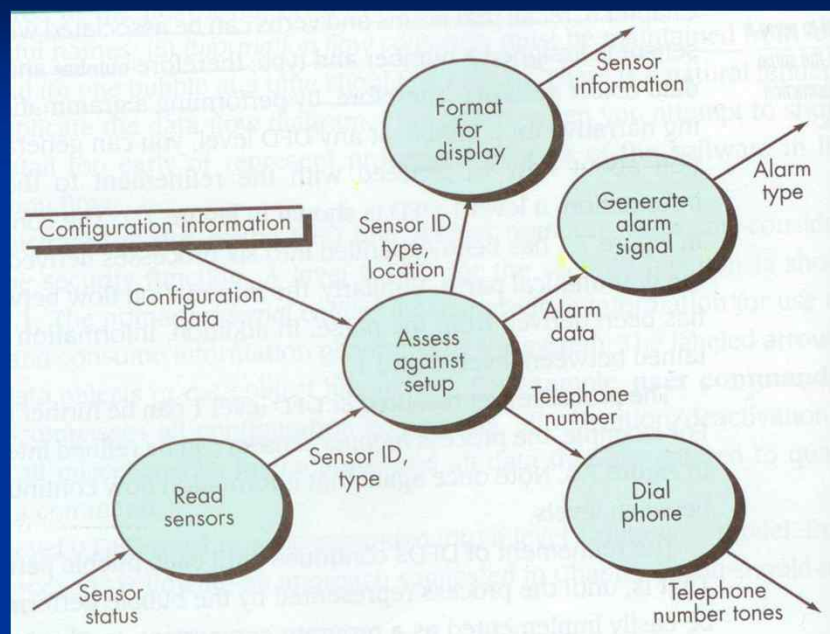
Data Storage

Flow-oriented Model

Flow-oriented Model
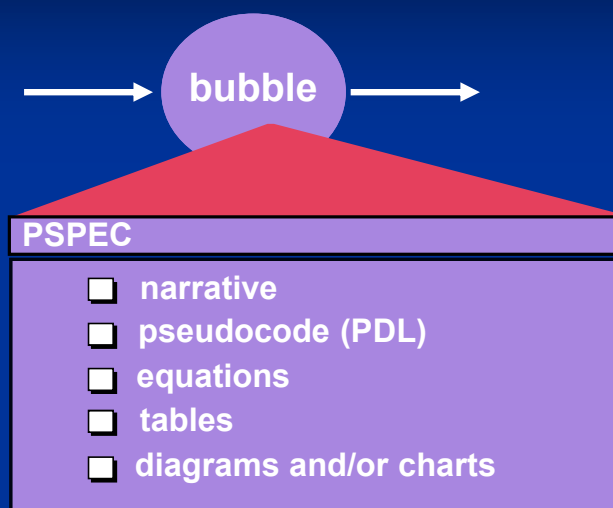

Flow-oriented Model

Flow-oriented Model

- The process specification (PSPEC)

  → Describe all flow model processes that appear at the <u>final level of refinement</u>

    - Narrative text
    - Program design language (PDL)
    - Mathematical equations
    - Tables
    - Diagrams
    - Chart

Process Specification (PSPEC)

Flow-oriented Model

| input events						
sensor event	0	0	0	0	1	0
blink flag	0	0	1	1	0	0
start stop switch	0	1	0	0	0	0
display action status complete	0	0	0	1	0	0
in-progress	0	0	1	0	0	0
time out	0	0	0	0	0	1
output						
alarm signal	0	0	0	0	1	0
process activation						
monitor and control system	0	1	0	0	1	1
activate/deactivate system	0	1	0	0	0	0
display messages and status	1	0	1	1	1	1
interact with user	1	0	0	1	0	1

Flow-oriented Model

- Creating a control flow model

  - Driven by <u>events rather than data</u>
  - Produce <u>control information</u>
  - Such applications require the use of <u>control flow modeling</u> in addition to data flow modeling

Flow-oriented Model

- The control specification

  - Represents the <u>behavior</u> of the system
  - <u>CSPEC contains a state diagram</u> that is a sequential specification of behavior

  (∗ CSPEC: Control Specification)

DFDs: A Look Ahead

analysis model

Maps into

design model
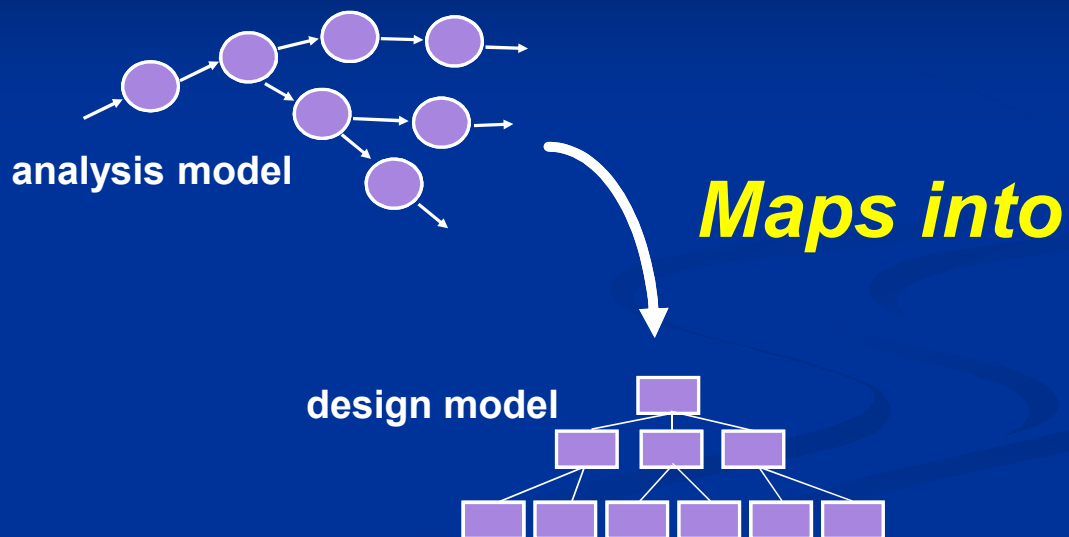
Behavioral Modeling

Behavioral Models

- **State diagrams**
- **Sequence diagrams**

Behavioral Modeling

- The behavioral model indicates how software will respond to external events or stimuli.

  1. Evaluate all use-cases
  2. Identify events that drive the interaction sequence and understand how these events relate to specific objects.
  3. A sequence for each use-case.
  4. A state diagram for the system.
  5. Review

Behavioral Modeling

- Identifying Events with the Use-Case

  - Safehome example: security function

    The homeowner used the keypad to key in a four-digit password. The password is compared with the valid password stored in the system. If the password is incorrect, the control panel will beep once and reset itself for additional input. If the password is correct, the control panel awaits further action
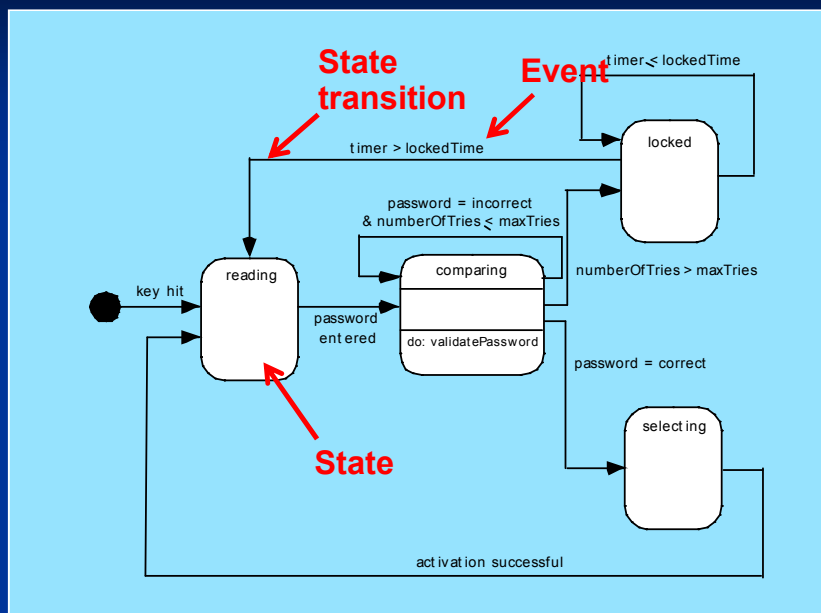
Creating a Behavioral Model

- State diagrams for analysis classes
  - UML state diagram that represents active states for each class and the events that cause changes between these active states

The States of a System

- State — a set of observable circumstances that characterizes the behavior of a system at a given time
- State transition — the movement from one state to another
- Event — an occurrence that causes the system to exhibit some predictable form of behavior
- Action — process that occurs as a consequence of making a transition

# State Diagram for the ControlPanel Class

Creating a Behavioral Model

- Sequence diagrams
    - Indicates how events cause transitions from object to object
    - The sequence diagram is s shorthand version of the use-case
    - Each of arrows represents an event and indicates how the event channels behavior between Safehome object
    - Time is measured vertically and the narrow vertical rectangles represent time spent in processing an activity
    - States may be shown along a vertical timeline

Sequence Diagram

Figure 8.27 Sequence diagram (partial) for *SafeHome* security function
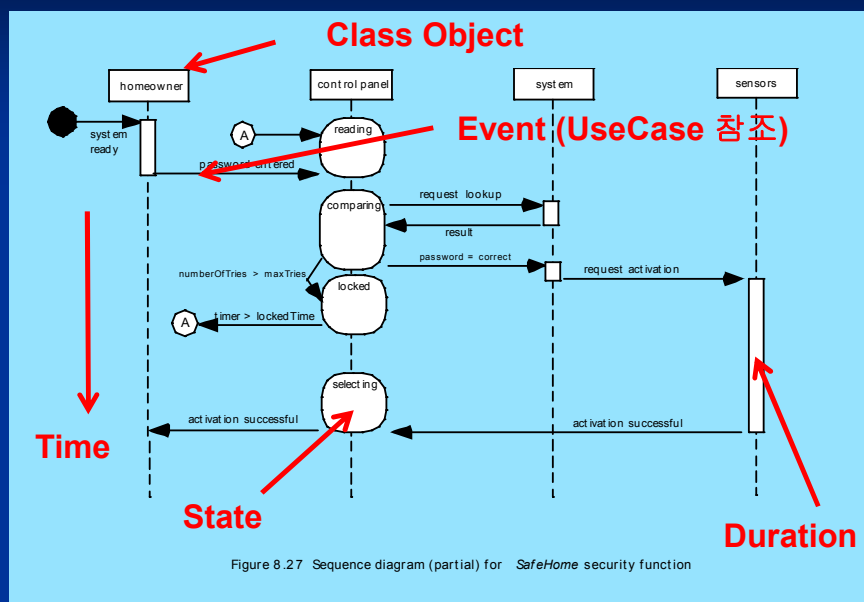
과제

- 이전 프로젝트 관리 과제에 대해 다섯 가지 분석 모델링을 적용하여 각각의 **diagram**을 그리고 간략히 설명하시오.