Slide 08 : Function-Oriented Software Design

Made by Md. Mehedi Hasan Rafy

Function-Oriented Software Design

- Views the system as a black box that provides high-level functions (services) to users.
- During design, these high-level functions are broken down step-by-step into detailed functions.

SA/SD Design Methodology

- ***SA (Structured Analysis):*** Converts *SRS* into a *Data Flow Diagram (DFD).*
- ***SD (Structured Design):*** Converts *DFD* into a *Structure Chart.*

Structured Analysis

- Converts *SRS* into a *Data Flow Diagram (DFD).*
- Analyzes high-level functions and shows data flow between them.

Key Principles:

1. Top-down decomposition
2. Divide and conquer – break each function into smaller parts
3. Use of graphical DFDs to show results

Data Flow Diagram (DFD)

- Also called a bubble chart.
- Shows input, processing (as bubbles), and output of a system.
- Focuses only on data flow, not execution order or control flow.
- Ignores algorithms and conditions for function execution.

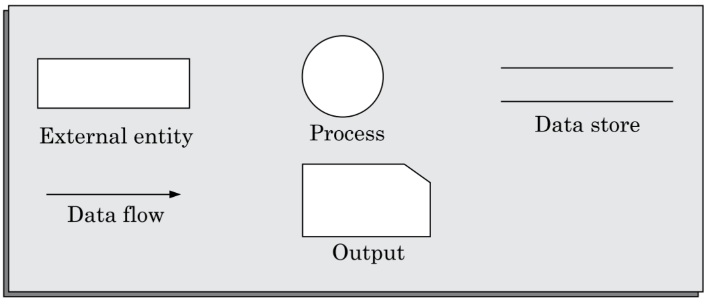
Difference from Flowchart:

- Flowchart shows control flow (sequence, branches).
- DFD shows data flow (no control or branches).

DFD Symbols

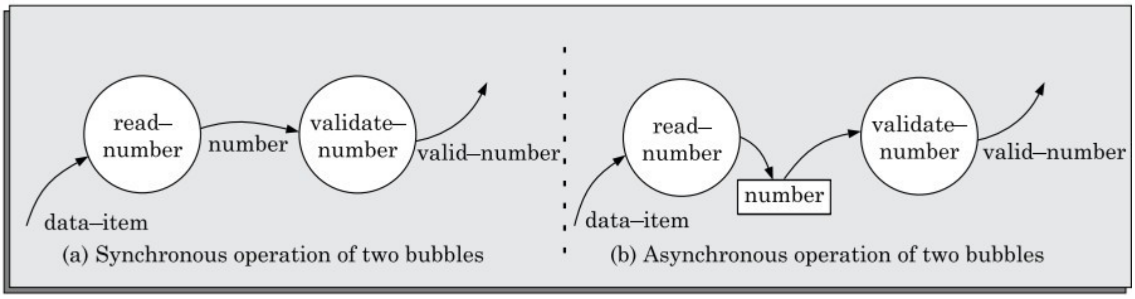
1. ***Process (Function/Bubble):***
    - Shown as a **circle**
    - Represents a **function/task**

2. ***External Entity:***
    - Shown as a **rectangle**
    - People, devices, or software **outside** the system



3. ***Data Flow:***
    - Shown as an **arrow**
    - Shows direction of **data movement**
4. ***Data Store:***
    - Shown as **two parallel lines**
    - Represents file or data structure (read/write shown by arrow direction)
5. ***Output Symbol:***
    - Used to show **printed or hard copy output**

Synchronous vs. Asynchronous Operations



- ***Synchronous:***
    - Two bubbles connected **directly** by a data flow
    - Work at the **same speed**
    - One waits for the other
- ***Asynchronous:***
    - Bubbles connected **via data store**
    - Work at **different speeds**
    - No need to wait

Importance of DFD

- Easy to understand and use.
- Represents system using hierarchical structure of functions.
- Helps the human mind grasp the system step by step – from simple to detailed.
- Useful not only in software design but also for showing document/item flow in organizations.

Data Dictionary

- A data dictionary lists all data items in a DFD (data flows + data stores).
- One dictionary covers all levels of the DFD model.

Importance:

1. Ensures standard terms for data — avoids confusion among developers.
2. Defines data structures clearly for implementation.
3. Helps in impact analysis – shows how data affects processes and vice versa.

Data Definition Operators (Used in Data Dictionary)

- `+` → **Composition :** `a + b` means both a and b appear
- `[,]` → **Selection :** `[a, b]` means either a **or** b appears
- `( )` → **Optional :** `a + (b)` means b may **or may not** appear
- `{ }` → **Iteration :** `{name}5` = 5 names, `{name}*` = 0 or more names
- `=` → **Equivalence :** `a = b + c` means a includes both b and c
- `/* */` → **Comment**