**Functional and Behavioural Modeling**

“What is the software supposed to do?”

**Lecture Objectives**

- To illustrate the modeling of functional and behavioural characteristics of the problem domain
- To describe the elements of data flow diagrams and the associated rules for drawing them correctly
- To describe the elements and usage of state transition diagrams

**Data Flow Diagram (DFD)**

- Graphical representation of functional modeling
- In analysis, provide representation of information flow in existing and required system
- In design, the DFDs can be decomposed into lower level processes (sub-systems) for implementation

**Context Diagram**

![Context Diagram](Image)

**Functional Modeling**

- In understanding the requirements of the software, the functions required by the customer will be identified
- All the functions process information in some way in the system
- Basically input → process → output
- Representation of how information is transformed

**Level-1 DFD**

![Level-1 DFD](Image)
**External Entity**

A producer or consumer of information that resides outside the bounds of the system to be modeled.

- **Source** - producer of information
- **Sink** - consumer of information

**Examples:**
- Customer
- Supplier
- Management

---

**Data Store**

A repository of data that is to be stored for use by one or more processes.

- May be as simple as a buffer or queue or as sophisticated as a relational database

**Examples:**

<table>
<thead>
<tr>
<th>Students</th>
<th>Inventory</th>
<th>Courses</th>
</tr>
</thead>
</table>

---

**Process**

A transformer of information (a function) that resides within the bounds of the system to be modeled.

**Examples:**

1. **Add New Member**
2. **Student Registration**
3. **Accept Registration Details**

---

**DFD Rules**

- No process can have only outputs.
- No process can have only inputs.
- A process has a verb label.

---

**Data Flow**

Data object that flows in the system; the arrowhead indicates the direction of data flow.

- new student record
- student record

---

**DFD Rules**

- Data cannot move directly from one data store to another data store.
- Data cannot move directly from an outside source to a data store.
- Data cannot move directly to an outside sink from a data store.
- A data store has a noun phrase label.
DFD Rules

Source/Sink
- Data cannot move directly from a source to a sink. It must be moved by a process if the data are of any concern to our system.
- A source/sink has a noun phrase label.

DFD Characteristics
- Can be used to model physical or logical, current or new systems
- Does not represent procedural or time-related processes
- Revisions to the same DFD are done to improve model based on understanding
- Decision to stop iterative decomposition may be difficult

DFD Rules

Data Flow
- A data flow has only one direction of flow between symbols.
- A fork in data flow means that exactly the same data goes from a common location to two or more processes/stores/sources/sinks.
- A join in a data flow means that exactly the same data comes from any two or more processes/stores/sources/sinks.

Behavoural Modeling

Representation of how the system changes and the events that cause the changes to happen
- Also represent actions that may be taken as consequences of events
- Graphically drawn as state transition diagram
State Transition Diagram (STD) Notation

State Transition Diagram (STD)

State Transition Diagram Elements

References


“Software Engineering” by Ian Sommerville, Addison-Wesley, 2001