Chapter 5
Data and Process Modeling Part One

Dr. Feng-Jen Yang
Topics

- Describe data and process modeling concepts and tools, including data flow diagrams, a data dictionary, and process descriptions
- Describe the symbols used in data flow diagrams and explain the rules for their use
- Draw data flow diagrams in a sequence, from general to specific
- Explain how to level and balance a set of data flow diagrams
Describe *how* a data dictionary is used and *what* it contains

Use *process description* tools, including structured English, decision tables, and decision trees

Describe the *relationship* between logical and physical models
Introduction

- In Chapters 5 & 6
  - Develop a *logical model* of the proposed system
  - Document the system requirements
- Logical model shows *what* the system must do
- Physical model describes *how* the system will be constructed
Overview of Data and Process Modeling Tools

- Systems analysts use many *graphical* techniques
  - To *describe* an information system
- A data flow diagram (DFD)
  - Use various *symbols* to show
    - The system transforms input data into useful information
Data Flow Diagrams

- Show how data *moves through* an information system but does *not* show program logic or processing steps
- A set of DFDs provides a logical model that shows *what* the system does, *not how* it does it
DFD Symbols

Gane and Sarson Symbols | Symbol Name | Yourdon Symbols
---|---|---
APPLY PAYMENT | Process | APPLY PAYMENT
BANK DEPOSIT | Data Flow | BANK DEPOSIT
STUDENTS | Data Store | STUDENTS
CUSTOMER | External Entity | CUSTOMER
DFD Symbols cont.

- Process symbol
  - Receive input data and produces output that has a different content, form, or both
  - Contain the *business logic*
    - Also called *business rules*
  - Referred to as a *black box*
Data flow symbol

- A *path* for data to move from one part of the information system to another

- A *line* with a single or double *arrowhead*

- Represent one or more *data items*
Spontaneous generation
- A process that produces output but has *no input* data flow

Black hole
- A process that has input but produces *no output*

Gray hole
- A process that has at least one input and one output but the input is *insufficient to generate* the output
Data store symbol
- Represent data that the system stores
- The physical characteristics of a data store are unimportant
  - Concern a *logical* model only
DFD Symbols cont.

- **Entity Symbol**
  - A shaded rectangle
  - The *name* of the entity appears inside the symbol
DFD Symbols cont.

- DFD shows only *external entities*
  - They *provide* data to the system or *receive* output from the system
- Example
  - A *customer* entity submits an order to an order processing system
  - A *patient* entity supplies data to a medical records system
External entities are also called *terminators*

- A entity that is a data *origin* or final *destination*
- Source
  - An entity that *supplies* data to the system
- Sink
  - An entity that *received* data from the system
### Correct and Incorrect Examples of Data Flows

<table>
<thead>
<tr>
<th>Type</th>
<th>Diagram</th>
<th>Correct/Incorrect</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process to Process</td>
<td><img src="image" alt="Process to Process" /></td>
<td>✔️</td>
<td>Good practice</td>
</tr>
<tr>
<td>Process to External Entity</td>
<td><img src="image" alt="Process to External Entity" /></td>
<td>✔️</td>
<td>Good practice</td>
</tr>
<tr>
<td>Process to Data Store</td>
<td><img src="image" alt="Process to Data Store" /></td>
<td>✔️</td>
<td>Good practice</td>
</tr>
<tr>
<td>External Entity to External Entity</td>
<td><img src="image" alt="External Entity to External Entity" /></td>
<td>✗</td>
<td>Incorrect practice</td>
</tr>
<tr>
<td>External Entity to Data Store</td>
<td><img src="image" alt="External Entity to Data Store" /></td>
<td>✗</td>
<td>Incorrect practice</td>
</tr>
<tr>
<td>Data Store to Data Store</td>
<td><img src="image" alt="Data Store to Data Store" /></td>
<td>✗</td>
<td>Incorrect practice</td>
</tr>
</tbody>
</table>
Creating a Set of DFDs

- Create a *graphical model* of the information system
  - Based on *fact-finding* results
- Three-step process
  - Step 1: Draw a *context* diagram
  - Step 2: Draw a *diagram 0* DFD
  - Step 3: Draw the *lower-level* diagrams
Creating a Set of DFDs cont.

Guidelines for drawing DFDs

- Draw the context diagram that fits on one page
- Use the name of the information system as the process name in the context diagram
- Use unique names within each set of symbols
- Do not cross lines
- Provide a unique name and reference number for each process
- Obtain as much user input and feedback as possible
Step 1: Draw a Context Diagram

- A context diagram
  - A top level view of an information system that shows the system’s boundaries and scope

- Start by
  - Placing a single process symbol in the center of the page
  - Identify it as process 0
Step 1: Draw a Context Diagram cont.

- Then
  - Placing the system entities around the perimeter of the page
  - Using data flows to connect the entities to the central process

- Data stores are *not shown* in the context diagram
  - Contained within the system and remain *hidden until* more detailed diagrams are created
Step 1: Draw a Context Diagram cont.
Step 2: Draw a Diagram 0 DFD

- **Zoom in** on the process 0
  - To show *major* internal processes, data flows, and data stores
- **Repeat** the *entities* and *data flows* that appear in the context diagram
  - To retain all connections that flow *into* and *out of* process 0
Step 2: Draw a Diagram 0 DFD cont.

- If same data flows in both directions
  - Can use a *double-headed* arrow

- While exploding a DFD
  - The higher-level diagram is called the *parent diagram*
  - The lower-level diagram is called the *child diagram*
  - *Functional primitive*
    - A process consisting of a single function that is *not exploded further*
Step 2: Draw a Diagram 0 DFD
Step 3: Draw the Lower-Level Diagrams

- Use leveling and balancing techniques
- Leveling
  - The process of *drawing* a series of increasingly detailed diagrams *until* all functional primitives are identified
  - Example
    - Use a series of *increasingly detailed* DFDs to describe an information system
      - Exploding, partitioning, or decomposing

- Balancing
  - Ensure that the input and output *data flows* of the *parent DFD* are maintained on the *child DFD*.

[Diagram of a flowchart showing processes involving customer orders, product detail verification, credit status, order acceptance, and picking details leading to warehouse operations.]
Data Dictionary

- A data dictionary, or data repository
  - A *central storehouse* of information about the system’s data
  - Define and describe all data elements and meaningful combinations of data elements
- An analyst uses the data dictionary
  - To collect, document, and organize specific facts about the system
Data Dictionary cont.

- Meaningful *combinations* of data elements
  - A data element
    - Also called a *data item* or *field*
    - The *smallest* piece of data that has *meaning*
  - Record
    - Also called *data structures*
    - A meaningful combination of related data elements
    - Included in a data flow or retained in a data store
Documenting the Data Elements cont.

- The more complex the system
  - The more difficult to maintain full and accurate documentation

- Modern CASE tools simplify the task
  - A CASE repository ensures data consistency
Documenting the Data Elements

- Must document *every data element* in the data dictionary
- The *objective*
  - To provide clear, comprehensive information about the data and processes that make up the system
The following attributes usually are recorded and described:

- Data element name or label
  - A meaning name to user
- Alias
  - Any name other than the standard data element name
- Type and length
- Default value
- Acceptable values
  - Domain and validity rules
Documenting the Data Elements cont.

- **Source**
  - The *origination point* for the data element

- **Security**
  - The individual or department that has *access* or *update privilege* for the data element

- **Responsible user(s)**
  - User(s) responsible for *entering* and *changing* values for the data element

- **Description and comments**
  - *Additional* notes
For the data element named SOCIAL SECURITY NUMBER