Systems Analysis and Design

PHASE 1
Systems Planning

PHASE 2
Systems Analysis

PHASE 3
Systems Design

PHASE 4
Systems Implementation

PHASE 5
Systems Operation & Support

Systems Analysis

Data and Process Modeling
Chapter 4

Data and Process Modeling
Objectives

- Describe data and process modeling concepts and tools
- Explain how structured analysis describes an information system
- Describe the symbols used in data flow diagrams and explain the rules for their use
- Explain the sequence of data flow diagrams, from general to specific
Objectives

- Explain how to level (分層) and balance (平衡) a set of data flow diagrams
- Draw a complete set of data flow diagrams for an information system
- Describe how a data dictionary is used and what it contains
Objectives

- Use process description tools, including structured English, decision tables, and decision trees
- Explain the interaction among data flow diagrams, the data dictionary, and process description
- Describe the relationship between logical and physical models
Introduction

- Systems analysis phase has three stages
  - Requirements determination (Chapter 3)
  - Requirements analysis (Chapters 4 & 5)
  - Evaluation of alternatives (Chapter 6)
Data Flow Diagrams

- Data flow diagrams (DFDs) show how data moves through an information system.
- DFDs represent a logical model that shows what a system does, not how it does it.
Data Flow Diagrams

- Data flow diagram symbols
  - Four basic symbols
    - Process 處理工作
    - Data flow 資料流
    - Data store 資料儲存
    - External entity 外部實體
  - Two popular symbol sets
    - Gane and Sarson
    - Yourdon
Data Flow Diagrams

- **Process symbol**
  - Symbol is a rectangle with rounded corners
  - Documented with process descriptions
  - Receive input data and produces output
  - Output has a different form, or content, or both
  - Details are shown in a process description
  - In DFDs the process symbol appears as a black box, underlying details not shown
Data Flow Diagrams

- Data flow symbol
  - Symbol is a line with an arrowhead showing direction
  - A path for data to move from one part of the system to another
  - Represents one or more data items
  - At least one data flow must enter and exit each process
Data Flow Diagrams

- Data flow symbol
  - Incorrect process and data flow combinations cause problems
    - Spontaneous generation (miracle)
    - Black hole
    - Gray hole
Data Flow Diagrams

- **Data store symbol**
  - Symbol is a rectangle open on the right side
  - Data store also is called a data repository
  - Represents data that is retained for later processing
  - Must be connected to a process with a data flow
  - Must have at least one outgoing and incoming data flow
Data Flow Diagrams

- **External entity symbol**
  - Symbol is a square, usually shaded
  - Represents a person, organization, or other system that provides data or receives output from the system
- **External entities are called terminators**
  - Source (supplies data to the system)
  - Sink (receives data from the system)

Click to see Figure 4-7

Click to see Figure 4-8
Data Flow Diagrams

- **External entity symbol**
  - Symbol is a square, usually shaded
  - Represents a person, organization, or other system that provides data or receives output from the system

- **External entities are called terminators**
  - Source (supplies data to the system)
  - Sink (receives data from the system)

- **Must follow specific rules for connecting DFD symbols**

Click to see Figure 4-9
Data Flow Diagrams

- **Context diagrams (全景圖)**
  - Top-level view that shows the systems’ boundaries scope
  - Represent the results of fact-finding
  - One process symbol, numbered 0 (zero) is drawn in the center
  - Data flows connect the process to the entities
Conventions for data flow diagrams

1. Each context diagram must fit on one page
2. Process name in the context diagram should be the name of the information system
3. Use unique names within each set of symbols
4. Do not cross lines. Use abbreviated identifications (# <9)
5. Use a unique reference number for each process symbol
Data Flow Diagrams

- **Diagram 0**
  - Displays more detail than the context diagram
  - Shows entities, major processes, data flows, and data stores

**TAKE A BREAK !!!**
Data Flow Diagrams

- Shows entities, major processes, data flows, and data stores
- Other characteristics
  - Can contain diverging data flows
  - Exploded (partitioned or decomposed) version of process 0
  - Diagram 0 is the child of the parent context diagram
  - Also can be called an overview or level 0 diagram
- Can contain functional primitives

Click to see Figure 4-14

Click to see Figure 4-15
Data Flow Diagrams

- Lower-level diagrams
  - Usually necessary to show more detail
Data Flow Diagrams

- Lower-level diagrams
  - Usually necessary to show more detail
- Design must consider
  - Leveling
  - Balancing
  - Data stores
Data Flow Diagrams

- Leveling
  - Process of drawing increasingly detailed diagrams
  - Also called exploding, partitioning, or decomposing

Click to see Figure 4-18
Data Flow Diagrams

- Balancing
  - Maintains consistency among an entire set of DFDs
  - Parent’s input and output data flows are preserved on the child

Click to see Figure 4-19
Data Flow Diagrams

- **Data stores**
  - Might not appear on higher-level DFDs
  - Are shown on the highest-level DFD that has two or more processes using that data store
Data Flow Diagrams

- Strategies for developing DFDs
  - Main objective is to ensure that your model is accurate and easy to understand
  - A diagram should have no more than nine process symbols
Data Dictionary

- Also called data repository
- Documents specific facts about the system
  - Data flows
  - Data stores
  - External entities
  - Processes
  - Data elements (data items, fields)
  - Records (data structures)
Data Dictionary

- Documenting the data elements
  - Must document every data element
  - Standard form or CASE tool can be used

Click to see Figure 4-25
Data Dictionary

- Documenting the data elements
  - Must document every data element
  - Standard form or CASE tool can be used
  - Various tools are available
    - *Visible Analyst* is a popular example
  - Key objective is to provide clear, comprehensive information about the system

Click to see Figure 4-26
Data Dictionary

❖ Documenting the data flows
  ❖ Must document every data flow
  ❖ Standard form or CASE tool can be used
  ❖ All major characteristics must be recorded and described

Click to see Figure 4-27
Data Dictionary

❖ Documenting the data stores
  ❖ Must document every data store
  ❖ Standard form or CASE tool can be used
  ❖ All major characteristics must be recorded and described
Data Dictionary

- Documenting the processes
  - Must document every process
  - Standard form or CASE tool can be used
  - All major characteristics must be recorded and described
Data Dictionary

- Documenting the external entities
  - Must document every external entity
  - Standard form or CASE tool can be used
  - All major characteristics must be recorded and described
Data Dictionary

- Documenting the records
  - Must document every record
  - Standard form or CASE tool can be used
  - All major characteristics must be recorded and described
Data Dictionary

- Data dictionary reports
  - Data dictionary serves as a central storehouse for documentation
  - Using this data, you can produce many valuable reports
Process Description Tools

- Modular design
  - Process description documents a functional primitive, using modular design
  - Modular design uses three logical structures
    - Sequence
    - Selection
    - Iteration

Click to see Figure 4-32

Click to see Figure 4-33

Click to see Figure 4-34
Process Description Tools

- **Structured English**
  - Subset of standard English
  - Describes process logic
    - Use only standard sequence, selection, and iteration structures
    - Use indentation for readability
    - Use a limited vocabulary
Process Description Tools

- **Decision tables**
  - Shows a logical structure that describes process logic
  - Every logical combination is shown initially
  - Results then can be combined and simplified
  - Programmers can use decision tables in developing code

Click to see Figure 4-37

Click to see Figure 4-38

Click to see Figure 4-39
Process Description Tools

- Decision trees
  - Graphical representation that shows a decision table’s conditions, actions, and rules
  - Logic structure is shown horizontally
  - Easy to construct and understand
  - Decision table is better in complex situations

Click to see Figure 4-40

Click to see Figure 4-41
Logical Versus Physical Models

- Sequence of models
  - A physical model shows how the systems’ requirements are implemented
  - Create a physical model of the current system
  - Develop a logical model of the current system
  - After the current system is understood, create a logical model of the new system
Logical Versus Physical Models

- Four-model approach
  - Four models
    - Physical model of the current system
    - Logical model of the current system
    - Logical model of the new system
    - Physical model of the new system
Logical Versus Physical Models

- Four-model approach
  - Major benefit is having a better grasp of the current system functions before making any modifications
  - Major disadvantage is added time and cost needed to develop a logical and physical model of the current system
The SWL team completed the fact-finding process

Rick and Carla are ready to prepare a logical model of the system
Data flow diagrams

Rick and Carla prepared a draft context diagram
Data flow diagrams
- Rick and Carla prepared a draft context diagram
- Various revisions resulted in final version
Data flow diagrams

Rick and Carla prepared a draft context diagram

Various revisions resulted in final version

Next steps

Analysts prepared diagram 0
SOFTWEAR, LIMITED

- Data flow diagrams
  - Rick and Carla prepared a draft context diagram
  - Various revisions resulted in final version

Next steps
- Analysts prepared diagram 0
- Rick partitioned the ESIP subsystem
- Carla developed other lower-level diagrams
- Logical model was completed
- Physical design issues were considered
SOFTWEAR, LIMITED

- Data dictionary and process descriptions
- Rick and Carla’s activities
  - Documented the ESIP subsystem
  - Met with Amy Calico to review the final model

Click to see Figure 4-47

Click to see Figure 4-48

Click to see Figure 4-49

Click to see Figure 4-50

Click to see Figure 4-51

Click to see Figure 4-52
Next steps
- Meet with SWL users to review the model
- Obtain input, make adjustments, get approval
- Complete the payroll system model
- Continue work on system requirements document
End

Chapter 4
FIGURE 4-1 Systems analysts often use visual aids during presentations.
**Figure 4-2** Data flow diagram symbols, symbol names, and examples from the Gane and Sarson and Yourdon symbol sets.
FIGURE 4-3  Examples of correct combinations of data flow and process symbols.
FIGURE 4-4

Examples of incorrect combinations of data flow and process symbols: APPLY INSURANCE PREMIUM has no inputs and is called a spontaneous generation process. CALCULATE GROSS PAY has no outputs and is called a black hole process. CALCULATE GRADE has an input that is obviously unable to produce the output. This process is called a gray hole.
FIGURE 4-5 Examples of correct uses of data store symbols in a data flow diagram.
FIGURE 4-6  Examples of incorrect uses of data store symbols: two data stores cannot be connected by a data flow without an intervening process, and each data store should have an outgoing and incoming data flow.
FIGURE 4-7 Examples of correct uses of external entities in a data flow diagram.
FIGURE 4-8  Examples of incorrect uses of external entity symbols. An external entity must be connected by a data flow to a process, and not directly to a data store or to another external entity.
## DATA FLOW THAT CONNECTS

<table>
<thead>
<tr>
<th>Connection</th>
<th>OKAY TO USE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>A process to another process</td>
<td>YES</td>
</tr>
<tr>
<td>A process to an external entity</td>
<td>YES</td>
</tr>
<tr>
<td>A process to a data store</td>
<td>YES</td>
</tr>
<tr>
<td>An entity to another entity</td>
<td>NO</td>
</tr>
<tr>
<td>An entity to a data store</td>
<td>NO</td>
</tr>
<tr>
<td>A data store to another data store</td>
<td>YES</td>
</tr>
</tbody>
</table>

**Figure 4-9** Rules for connecting processes, data stores, and external entities in a DFD.
FIGURE 4-10 Context diagram DFD for the grading system.
FIGURE 4-11  Context diagram DFD for an order system.
FIGURE 4-13  Context diagram and diagram 0 for the grading system.