System Analysis (Phase-2) – Part 2 Data and Processing Modeling

Peter Lo

Objective

- Data Flow Diagrams
- Data Dictionary
- Process Description Tools
- Logical vs. Physical Model

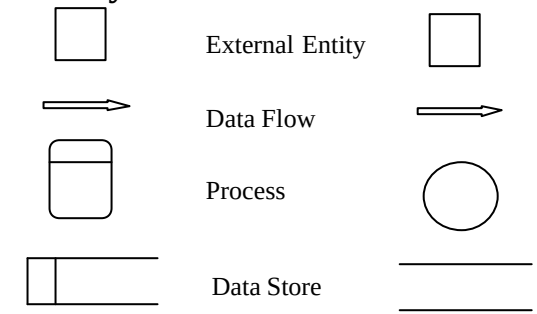
Data Flow Diagrams (DFD)

- A Data Flow Diagram (DFD) is a graphical tool to depict the flow of data through a system and the work or processing performed by that system.
  - It shows how data moves through an information system but does not show program logic or processing steps.
  - It only represent a logical model that shows what the system does, not how it does it.

Why DFD ?

- Language Description is subject to interpretation, it may omit crucial info.
- Graphical Description of the flow of data within an organization with DFD.

DFD Symbols/Elements



Gane & Sarson Notations

De Marco & Yourdon Notations

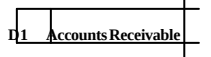
DFD Symbols – Process

- A Process is a work or action performed on input data flow to produce an output data flow
- Use a verb to label the action performed by the process (not the name of person or department who does it as in physical DFD)
- A Process must have at least one input data flow and at least one output data flow.
- Examples: Apply rent payment, verify order, pay bill.

DFD Symbols – Data Flows

- A Data Flow represents a movement of data (information) among processes or data stores
- A Data Flow does not represent a document or a physical good: it represents the exchange of information in the document or about the good
- A Data Flow represents an input of data to a process, or the output of data from a process.
  - A data flow may also be used to represent the creation, reading, deletion, or updating of data in a file or database (data store).
  - A composite data flow (packet) is a data flow that consists of other data flows.
- Examples: Deposit, invoice payment, delivery slip.

DFD Symbols – Data Store



- A Data Store is a storage of data: it contains information
- Physical storage is immaterial: it can be a filing cabinet, book, computer file
- A data store is an inventory of data.
  - A data store is "data at rest" compared to a data flow that is " data in motion."
  - Almost always one of the following: Persons (or groups of persons), Places, Objects, Events (about which data is captured), Concepts (about which data is important),
  - One can identify data stores with REAL framework
  - Data stores depicted on a DFD store all instances of data entities (depicted on an ERD)
- Examples: Accounts receivable.

DFD Symbols – External Entity

- An External Entity is a provider (source) or receiver (sink) of data and info of the system
- An External Entity is not part of the system: the externality depends on how the system is defined
- An external entity (agent) defines a person, organization unit, or other organization that lies outside of the scope of the project but that interacts with the system being studied.
  - External agents define the "boundary" or scope of a system being modeled.
  - As scope changes, external agents can become processes, and vice versa.
  - Almost always one of the following: Office, department, division inside the business but outside the system scope, an external organization or agency, another business or another information system, one of system's end-users or managers
- Examples: Customer, student, supplier.

Context Diagram

- An overview of an organizational system that shows the system boundaries, external entities that interact with the system, and the major information flows between the entities and the system.
- Single process (labeled "0") represents the entire system. The sources/sinks represent its environmental boundaries.
- No data stores appear on context diagram.

Strategies for Developing DFD

- Top-down Model
  - First create the context diagram, then diagram 1, then all the child diagrams for diagram 1, and so on.
- Bottom-up strategy
  - First identify all functional primitives, data stores, external entities, and data flows. Then group processes with other related symbols to develop the lowest-level diagrams. Next, group those diagrams in a logical way to form the next higher level. Continue to work upwards until you reach Context Level.

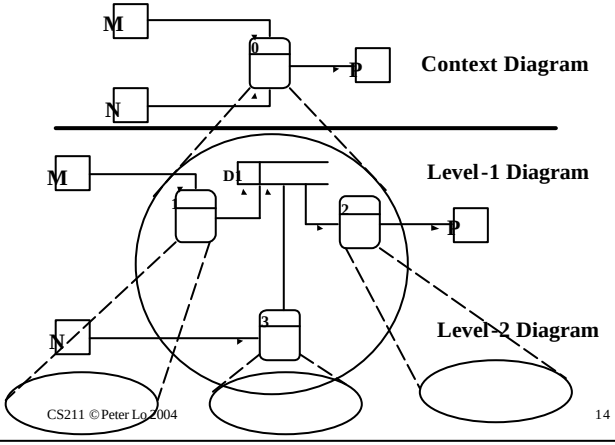
DFD Building Procedure

- Step 1: Context Diagram
  - Identify the system and its boundaries (the context)
  - Identify external entities (providers, receivers of system info)
  - Identify external data flows (input, output)
- Step 2: Level-1 DFD
  - Identify what is being done between each input and its corresponding output
  - Identify the processes
  - Identify external data flows between external entities and processes
  - Identify internal data flows between processes and data stores
- Step 3: Level-2 DFD's
  - Sub-processes (primitive processes) of Level-1 processes

Lower-Level Diagrams

- When lower-level diagrams are needed to show detail, it is essential that they be leveled and balanced.
- **leveling** is the process of drawing a series of increasingly detailed diagrams, until the desired degree of detail is reached.
- **Balancing** maintains consistency among the entire series of diagrams, including input and output data flows, data definition, and process descriptions.
- A balanced set of DFD preserves the input and output data flows of the parent on the child DFD.

Decomposition of Context Diagram

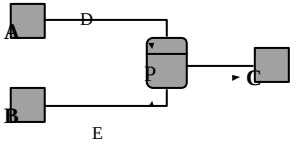


Data Flow Diagramming Rules

- There are two DFD guidelines that apply most of the time:
  - The inputs to a process are different from the outputs of that process.
  - Objects on a DFD have unique names.
- The following rules allow you (or a CASE tool) to evaluate DFDs for correctness.

Data Flow Diagramming Rules

- **Rule 1**: Unique label for each symbol to avoid confusion

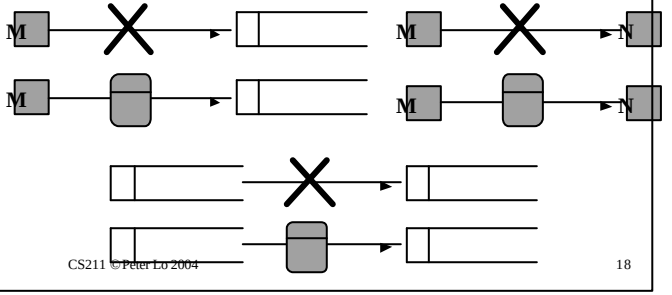


Data Flow Diagramming Rules

- **Rule 2**: Use an action VERB to label a process

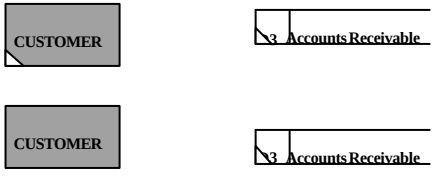
Data Flow Diagramming Rules

- **Rule 3**: Must be one process associated with each data flow.



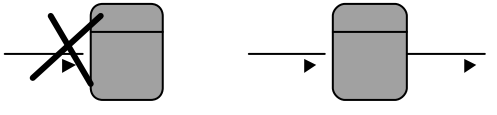
Data Flow Diagramming Rules

- **Rule 4**: Shaded corner must appear in ALL occurrences of a duplicated symbol in a same diagram.



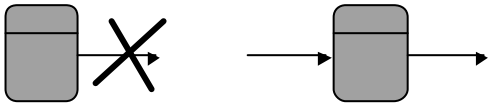
Data Flow Diagramming Rules

- **Rule 5**: No process without output data flow.



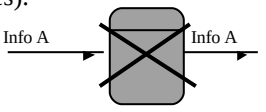
Data Flow Diagramming Rules

- **Rule 6**: No process without input data flow.



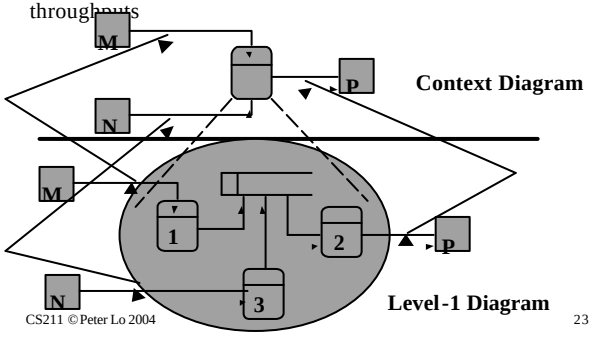
Data Flow Diagramming Rules

- **Rule 7**: No need for routing (without transforming) a data flow with a process (non value-added activities).



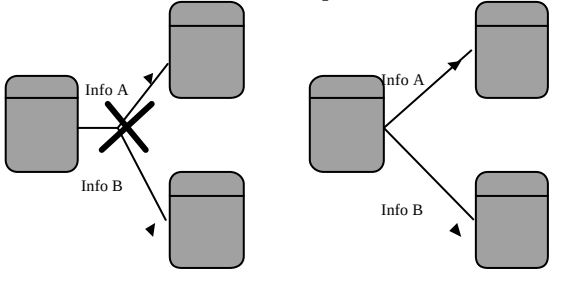
Data Flow Diagramming Rules

- **Rule 8**: Identical input, output data flows for parent and child processes (but the child processes can have their own throughputs



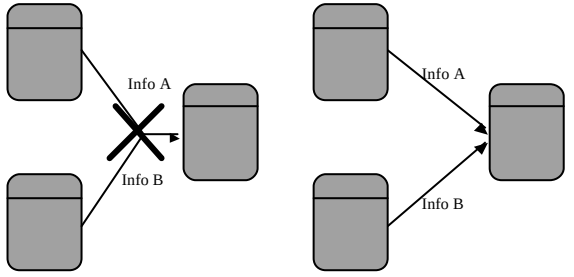
Data Flow Diagramming Rules

- **Rule 9**: Data flows cannot split



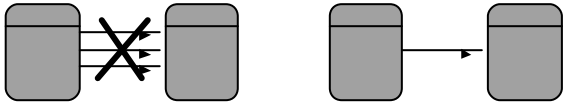
Data Flow Diagramming Rules

- **Rule 10**: Data flows cannot combine



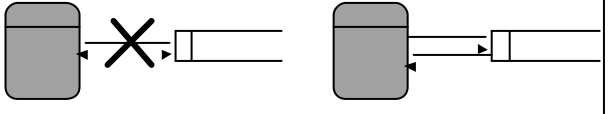
Data Flow Diagramming Rules

- **Rule 11**: A data packet can combine many data elements being transmitted at the same time to the same destination



Data Flow Diagramming Rules

- **Rule 12**: Double-headed arrows are forbidden [in-flow (update) and out-flow (extract info) of a data store are different]



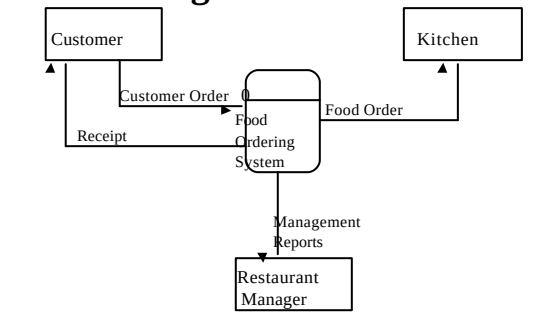
Conventions for DFD

- Conventions, or rules you should use when constructing DFD:
  - Each context diagram must fit on one page.
  - The process name in the context diagram should be the name of the information system.
  - Use unique names within each set of symbols.
  - Do not cross lines.
  - Use a unique reference number for each process symbol.

Example

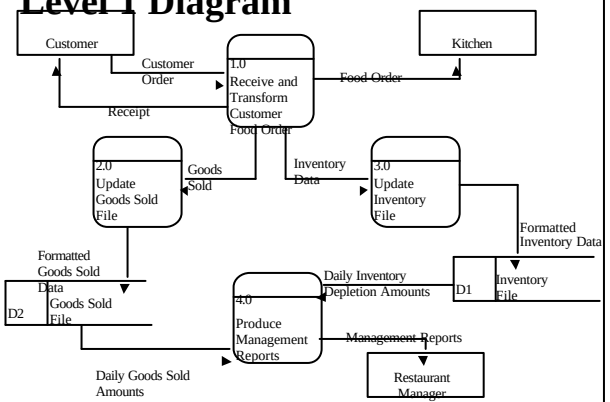
- Hoosier Burger uses an Information System to take customer orders, send the orders to the kitchen, monitor goods sold and inventory, and generate reports for management.
- The context DFD for the system is given in next page.
- The next step for the analyst to think about which processes are represented by the single process in the context diagram. 4 separate processes identified. These processes represent the major functions of the system.
- Level-1 diagram represents the primary individual processes in the system at the highest possible level. Each process has a number which ends in .0 (corresponding to the level number of the DFD).

Context Diagram

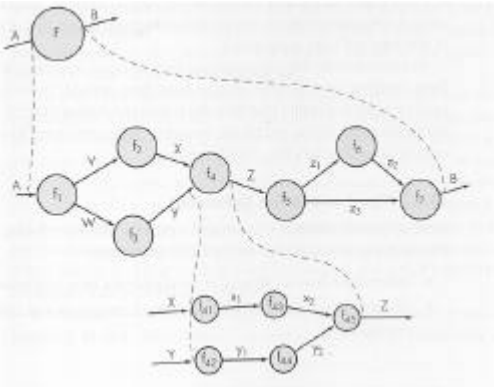


Context DFD of Hoosier Burger's food ordering system

Level 1 Diagram



Data Flow Diagrams (Example)



Data Dictionary

- A **Data Dictionary** is a central storehouse of information about the system's data.
- A **Data Element**, or **Data Item** or **Field**, is the smallest piece of data that has meaning within an information system.
- A **Record** is a meaningful combination of related data elements that is included in a data flow or retained in a data store.
- Contents of the data dictionary, includes data flows, data stores, data structures and records, data elements, external entities, and processes.

Documenting the Data Elements

- You must document every data element in the data dictionary. The following attributes are recorded in the data dictionary:
  - Data element name or label
  - Alternate name(s)
  - Type and length
  - Output format
  - Default value
  - Prompt, column header, or field
  - Caption
  - Source
  - Security
  - Responsible user(s)
  - Acceptable values and data validation
  - Derivation formula
  - Description or comments

Documenting the Data Flows

- Document all data flows in the data dictionary. The typical attributes are as follows:
  - Data flow name or label
  - Alternate name(s)
  - Description
  - Origin
  - Destination
  - Record
  - Volume and frequency

Documenting the Data Stores

- You must document every DFD data store in the data dictionary. Typical characteristics of a data store are as follows:
  - Data store name or label
  - Alternate name(s)
  - Description
  - Input data flows
  - Output data flows
  - Record
  - Volume and frequency

Documenting the Processes

- You must document every process. Following are typical characteristics of a process:
  - Process name or label
  - Purpose or description
  - Process number
  - Input data flows
  - Output data flows
  - Process description

Documenting the External Entities

- By documenting all external entities, the data dictionary can serve as a complete documentation package. Typical characteristics of an external entity include the following.
  - External entity name
  - Alternate name(s)
  - Description
  - Input data flows
  - Output data flows

Documenting the Records

- A record is a data structure that contains a set of related data elements that are stored and processed together.
- Data flows and data stores consist of records that you must document in the data dictionary. Typical characteristics of a record include the following:
  - Record or data structure
  - Alternate name(s)
  - Definition or description
  - Record content or composition

Data Dictionary Reports

- The Data Dictionary serves as the central storehouse of documentation for an information system.
- You can obtain many valuable reports from a data dictionary, including the following:
  - An alphabetized list of all data elements by name
  - A report by user departments of data elements that must be updated by each department
  - A report of all data flows and data stores that use a particular data element
  - Detailed reports showing all characteristics of data.

Example


Date: 12/19/2003 Project: WEB LANE CAFE Page: 11
Time: 10:35:28 AM
Detailed Listing – Alphabetically
All Entries – Data Flow Diagrams
Vendor-ID Data Element
Vendor File: Vendor-ID
Description:
A unique identification number assigned to each vendor.
Alias:
Vendor Code
Values & Meanings:
Required element
Cannot be blank
May not be duplicated
Data element attributes:
Storage Type: Char
Length: 4
Display Format: AAAA
Null Type: Not Null
Location:
File --> Vendor File
Date Last Altered: 12/19/2003 Date Created: 12/19/2003

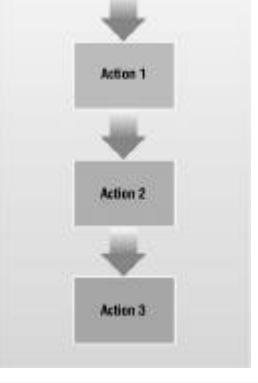
Process Description Tools

- A process description documents the details of a functional primitive, which represents a specific set of processing steps and business logic.
- Typical process description tools include:
  - Structured English
  - Decision tables
  - Decision trees

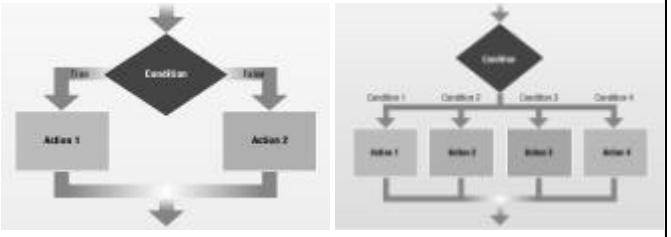
Modular Design

- **Modular Design** is based on combinations of three logical structures.
- The three structures are called sequence, selection, and iteration which using 3 symbols to combine.
  - **Rectangle** represents a step or process
  - **Diamond** shape represents a condition or decision
  - **Arrows** indicates the flow of logic
- Sequence, selection, and iteration structures can be combined in various ways to describe processing logic.

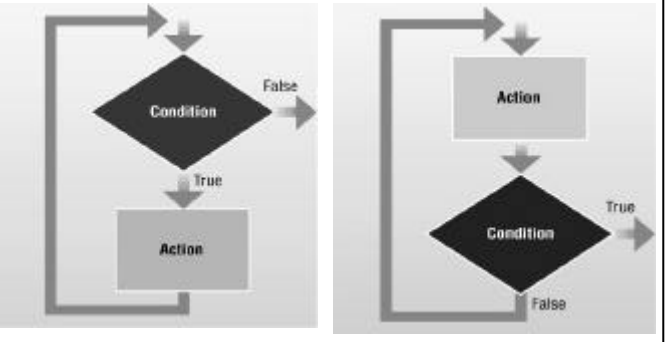
Modular Design – Sequence Structure



Modular Design – Selection Structure



Modular Design – Iteration Structure



Structured English

- **Structured English** is a subset of standard English that describes logical processes clearly and accurately.

UPLOADING VENDOR INFORMATION

For each item containing vendor information, perform the following steps:
    If the item is not a computer file then
        Use the scanner to convert it into a file format.
    Copy the file into the Vendor Information folder on your hard disk.
Zip all new files in the Vendor Information folder into a single file.
Save the zipped file in a Web folder.
E-mail the Webmaster with the name of the zipped file.

Structured English

- When you use structured English conform to the following rules:
  - Use only the three building blocks of sequence, selection, and iteration.
  - Use indentation for readability.
  - Use a limited vocabulary, including standard terms used in the data dictionary and specific words that describe the processing rules.

Decision Tables

- A decision table shows a logical structure, with all possible combinations of conditions and resulting actions.

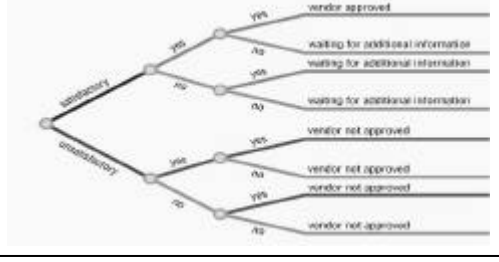
		1	2	3	4	5	6	7	8
CONDITIONS	Background check results (S = Satisfactory, U = Unsatisfactory)	S	S	S	S	U	U	U	U
	References furnished?	Y	Y	N	N	Y	Y	N	N
	Passed credit check?	Y	N	Y	N	Y	N	Y	N
ACTIONS	Vendor approved	X							
	Vendor not approved					X	X	X	X
	Waiting for additional information		X	X	X				

Decision Tables

- To create a decision table, follow these steps:
  - Place a heading at the top left that names the table.
  - Enter the conditions under the heading, with one condition per line.
  - Enter all potential combinations of Y/N (for yes and no) for the conditions. Each column represents a numbered possibility called a rule.
  - Place an X in the action entries area for each rule to indicate whether to accept or reject the order.

Decision Trees

- A decision tree is a graphical representation of the conditions, actions, and rules found in a decision table.



Decision Trees

- A decision tree is read from left to right, with the conditions along the various branches and the actions at the far right.

Logical Versus Physical Models

- Sequence of Models
  - First study the physical operations of the existing system to understand how the current tasks were carried out.
  - Develop a logical model of the current system
  - Develop the logical model of the new system.
- Performing that extra step allows analyst to understand the current system better.

Logical Versus Physical Models

- Four-Model Approach
  - Following this sequence in developing an information system, there will be a total of 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