US-SIT-3P2 Software Engineering

Data Flow Diagrams

Practical 3

S.Y. B.Sc IT : Semester 3

Data Flow Diagrams

A structured analysis technique that employs a set of visual representations of the data that moves through the organization, the paths through which the data moves, and the processes that produce, use, and transform data.

Why Data Flow Diagrams?

- Can diagram the **organization** or the **system**
- Can diagram the **current** or **proposed** situation
- Can facilitate **analysis** or **design**
- Provides a good bridge from analysis to design
- Facilitates communication with the user at all stages

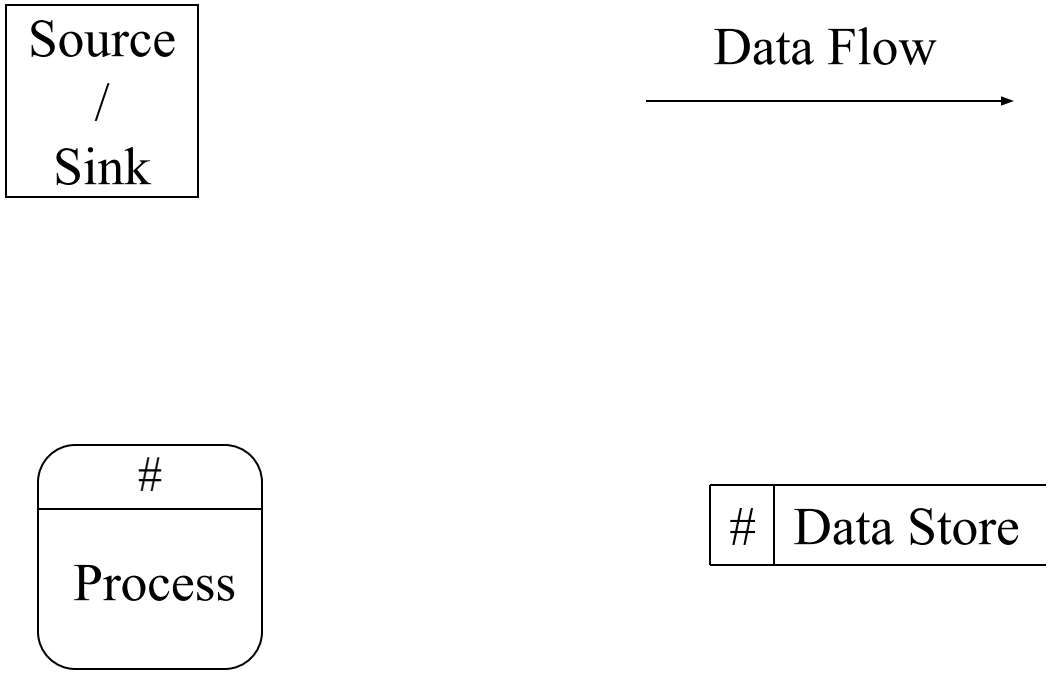
Types of DFDs

- **Current** - how data flows now
- **Proposed** - how we'd like it to flow
- **Logical** - the "essence" of a process
- **Physical** - the implementation of a process
- **Partitioned physical** - system architecture or high-level design

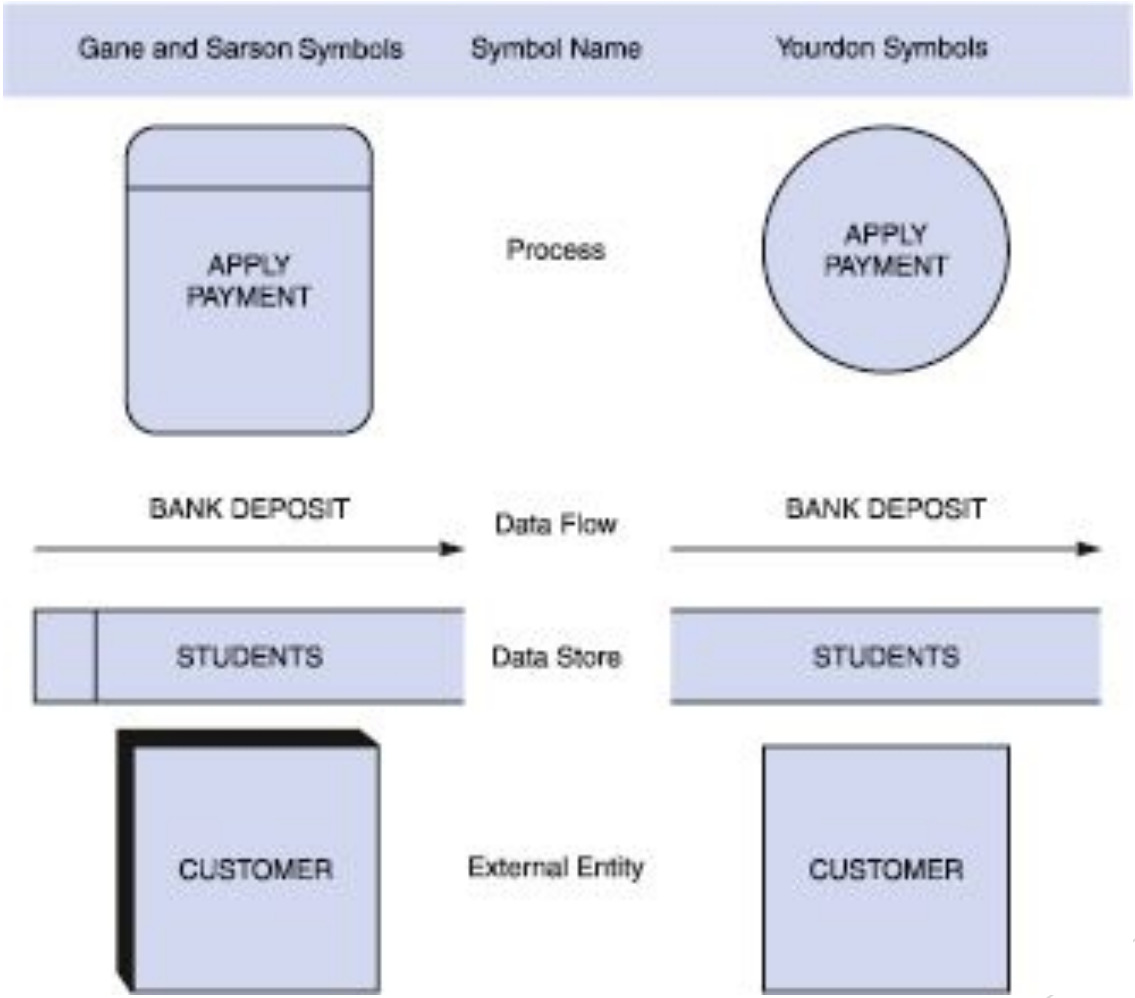
Levels of Detail

- **Context level diagram** - shows just the inputs and outputs of the system
- **Level 0 diagram** - decomposes the process into the major subprocesses and identifies what data flows between them
- **Child diagrams** - increasing levels of detail
- **Primitive diagrams** - lowest level of decomposition

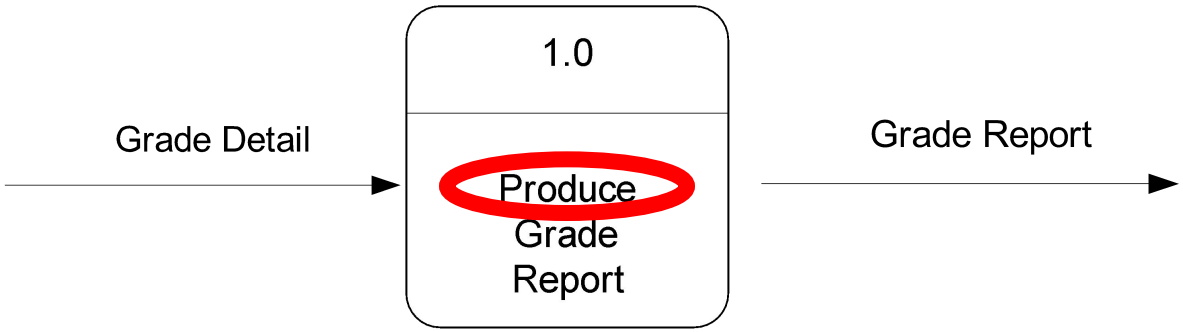
Four Basic Symbols



DFD Symbols

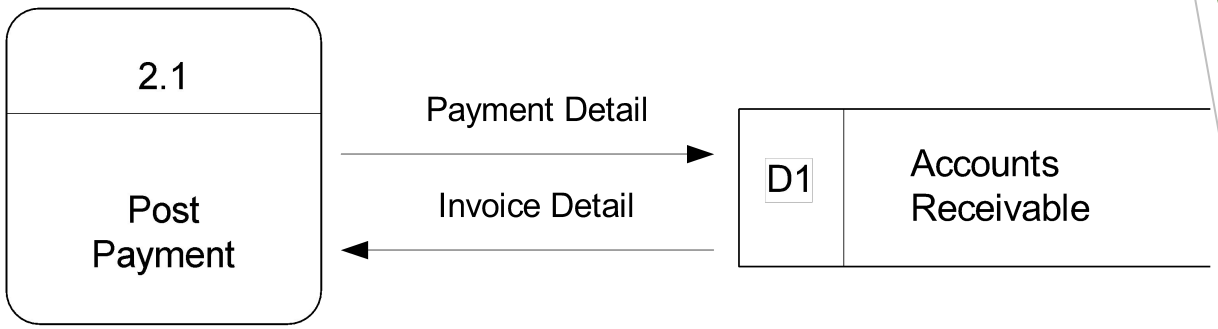

Gane and Sarson Symbols
Symbol Name
Yourdon Symbols
APPLY PAYMENT
Process
APPLY PAYMENT
BANK DEPOSIT
Data Flow
BANK DEPOSIT
STUDENTS
Data Store
STUDENTS
CUSTOMER
External Entity
CUSTOMER

Process



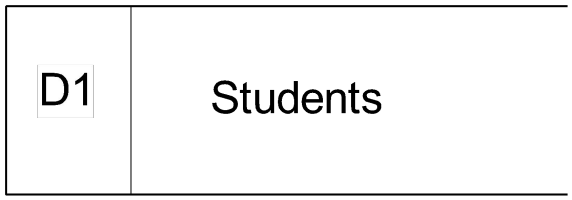
- The work or actions performed on data so that they are transformed, stored, or distributed.
- Process labels should be **verb phrases!**

Data Flow



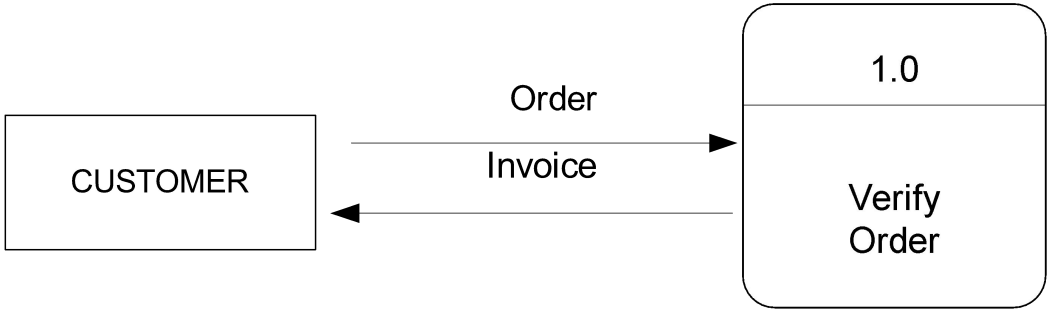
- A path for data to move from one part of the system to another.
- Data in motion!
  - Arrows depict the movement of data.
- **NO VERBS**

Data Store



- Used in a DFD to represent data that the system stores
- Data at rest!
- Labels should be noun phrases
  - **(NO VERBS)**

External Entity or Source/Sink



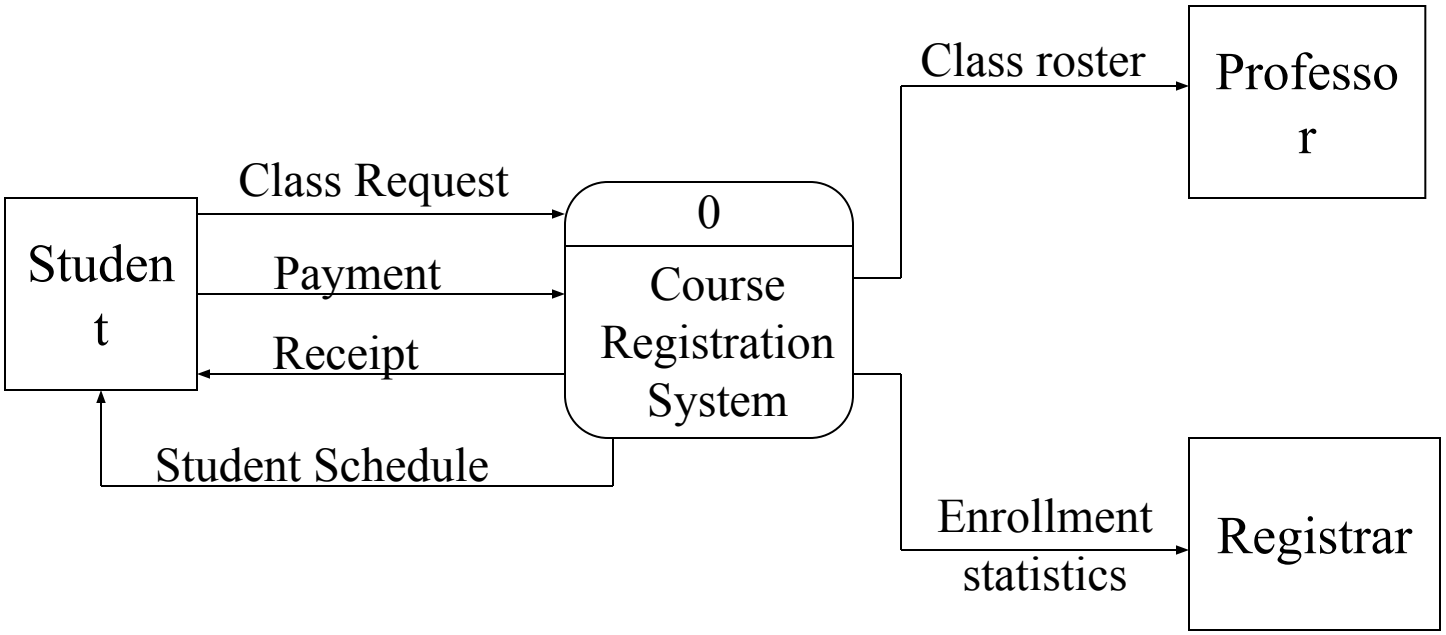
- The origin or destination of data!
  - This represents things outside of the system.
- Source – Entity that supplies data to the system.
- Sink – Entity that receives data from the system.
- The labels should be noun phrases!

Context Level Diagram

- Just one process
- All sources and sinks that provide data to or receive data from the process
- Major data flows between the process and all sources/sinks
- No data stores

Running Example

Course Registration: Context level Diagram

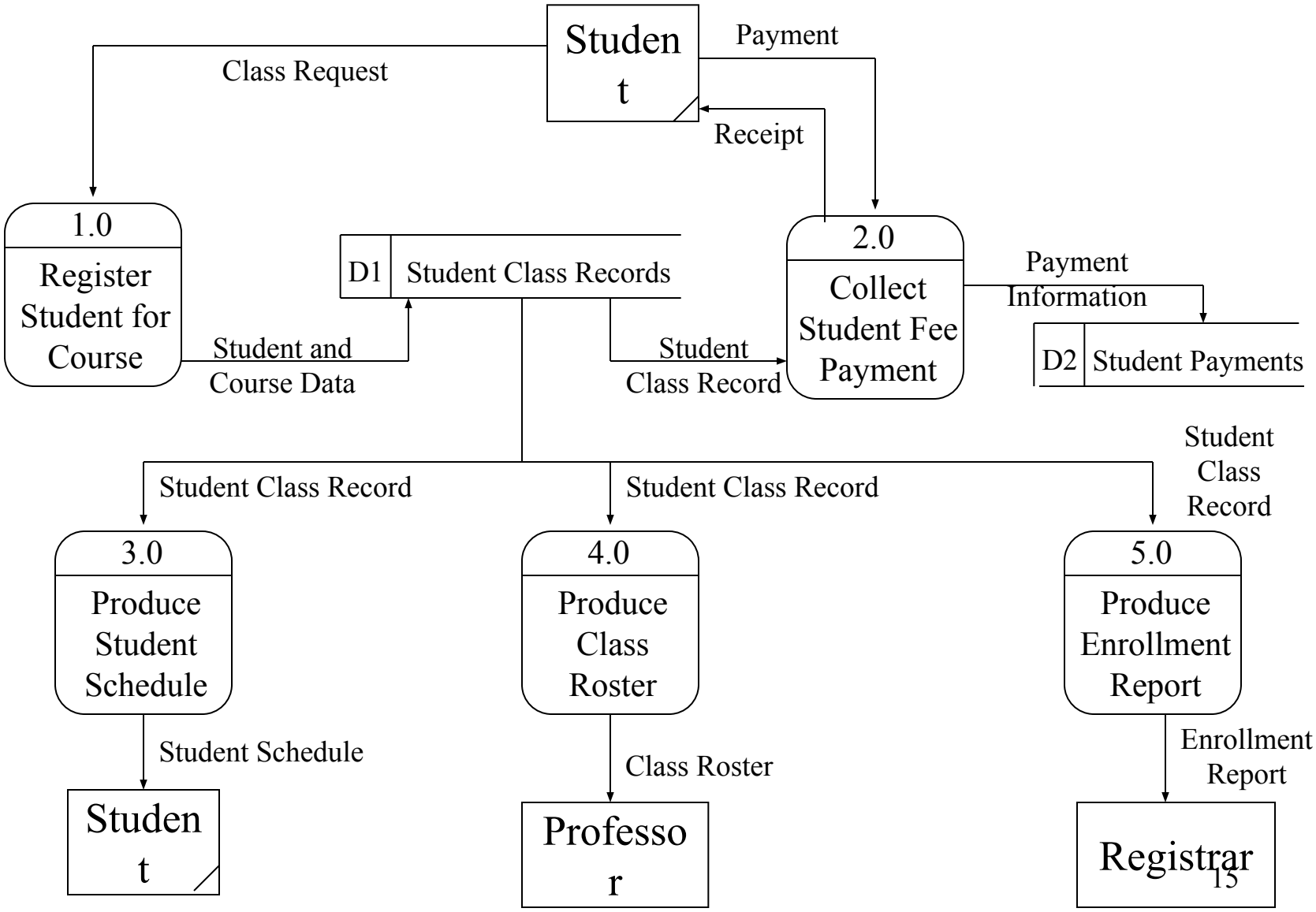


Level 0 Diagram

- Process is “exploded”
- Sources, sinks, and data flows repeated from context diagram
- Process broken down into subprocesses, numbered sequentially
- Lower-level data flows and data stores added

Running Example

Course Registration: Current Logical Level 0 Diagram

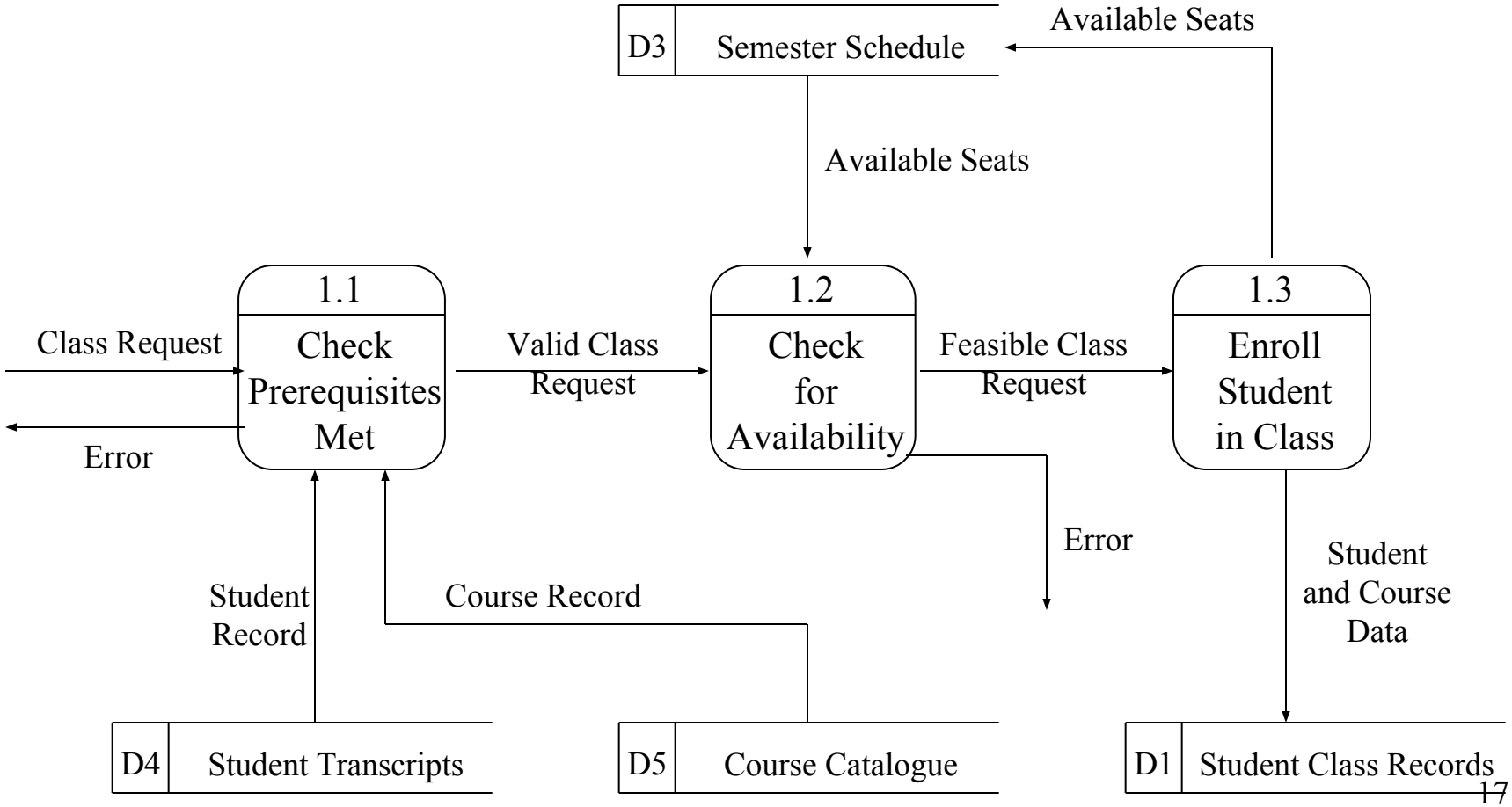


Child Diagrams

- “Explode” one process in level 0 diagram
- Break down into lower-level processes, using numbering scheme
- Must include all data flow into and out of “parent” process in level 0 diagram
- Don’t include sources and sinks
- May add lower-level data flows and data stores

Running Example

Course Registration: Current Logical Child Diagram

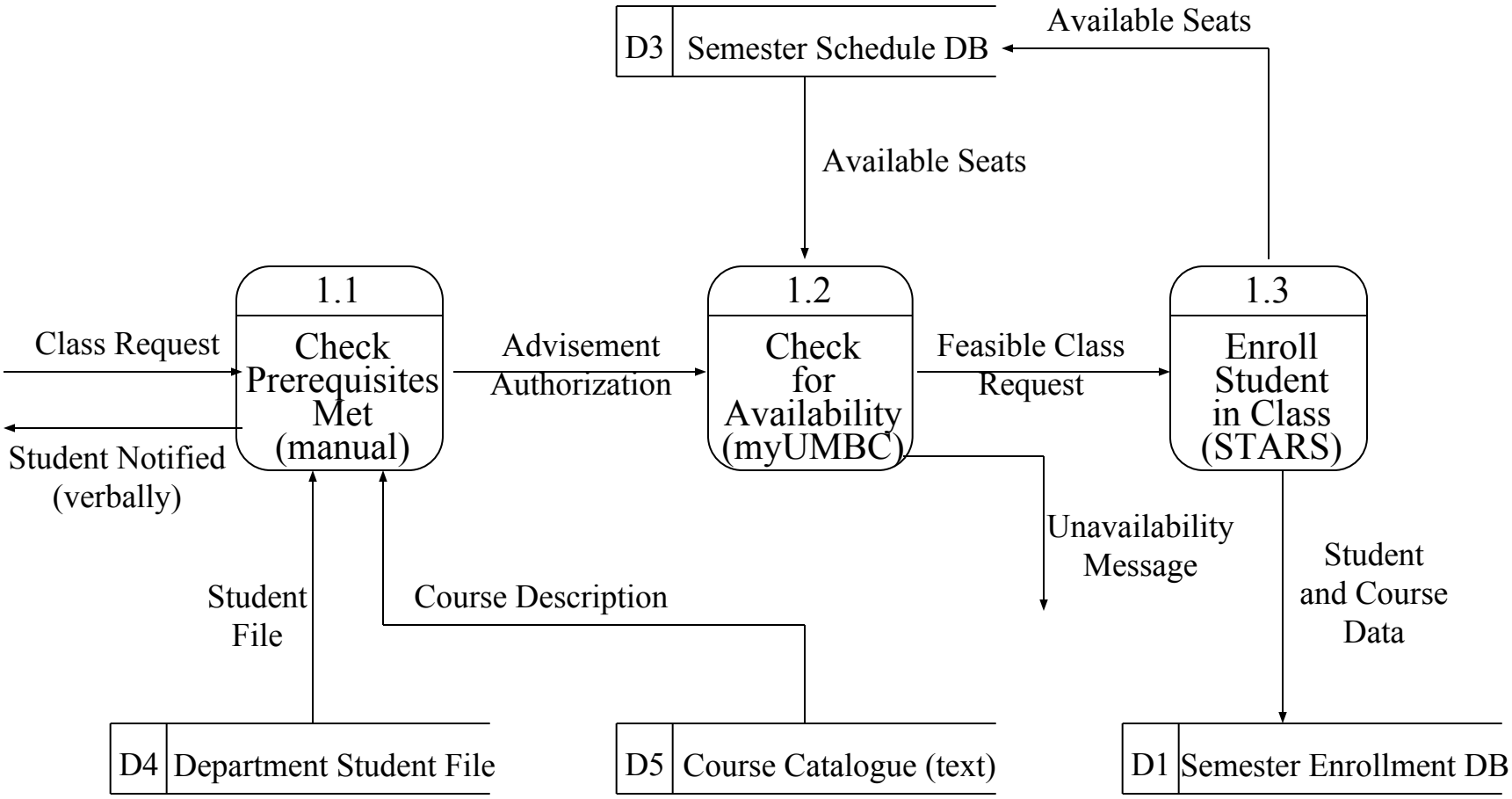


Physical DFDs

- Model the implementation of the system
- Start with a set of child diagrams or with level 0 diagram
- Add implementation details
  - indicate manual vs. automated processes
  - describe form of data stores and data flows
  - extra processes for maintaining data

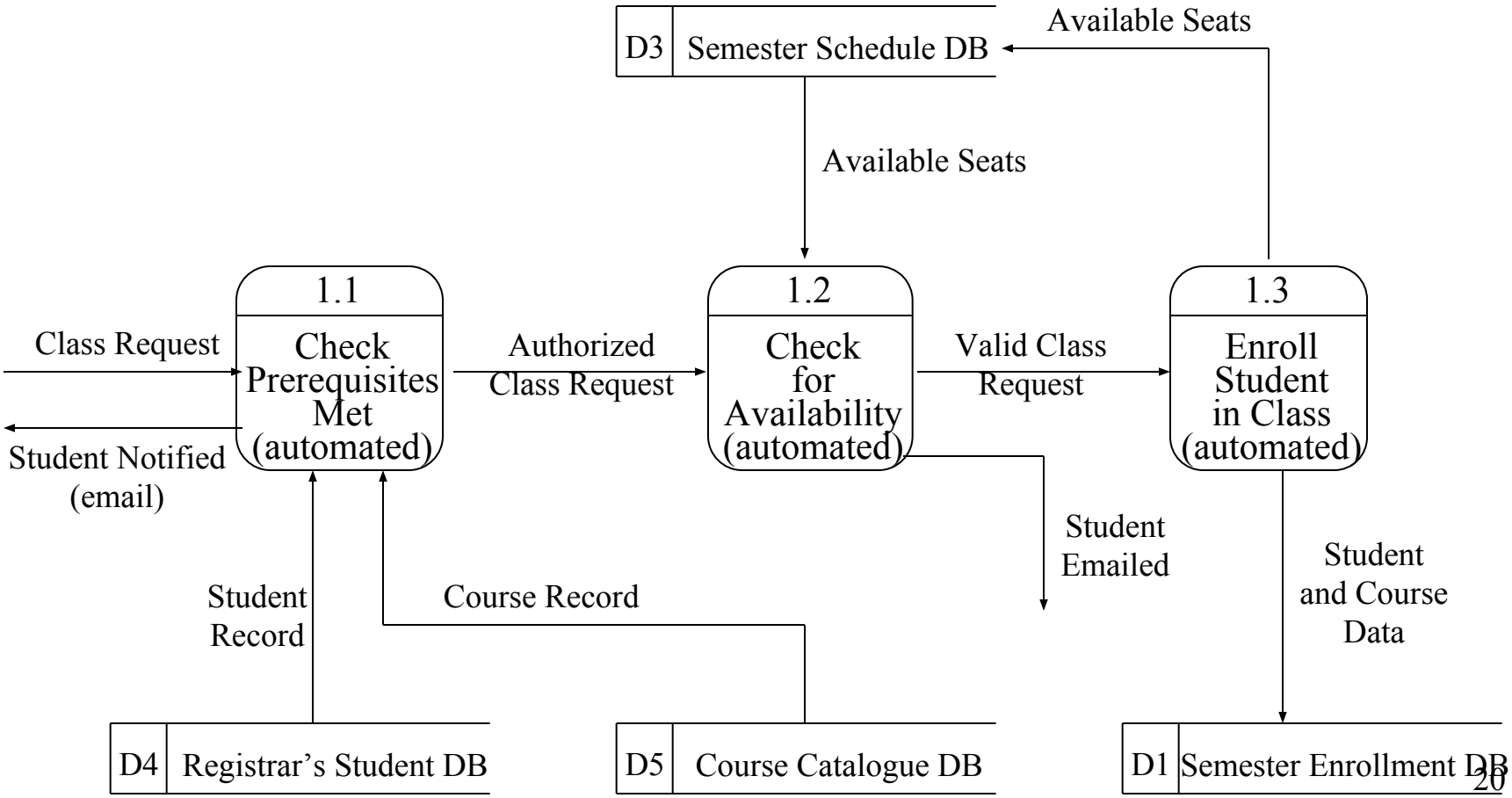
Running Example

Course Registration: Current Physical Child Diagram



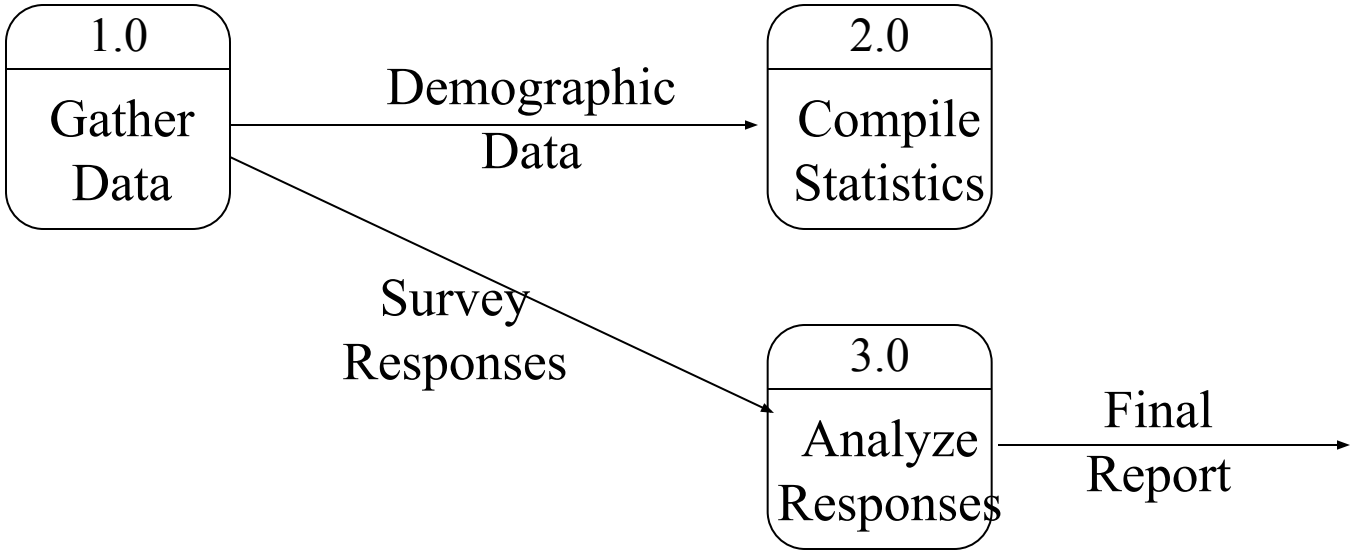
Running Example

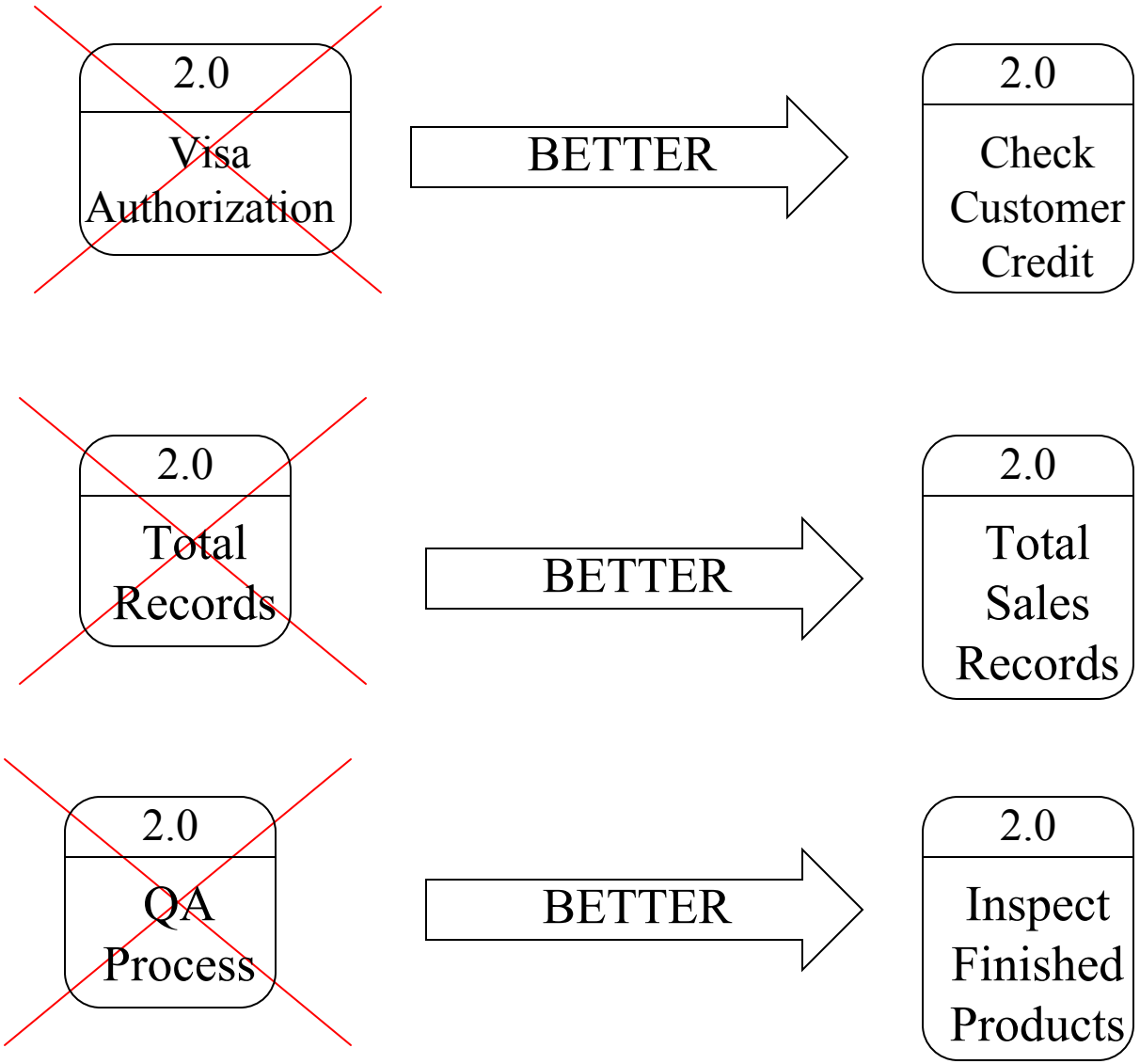
Course Registration: Proposed Physical Child Diagram



Data Flow Diagramming Rules

- Processes
  - a process must have at least one input
  - a process must have at least one output
  - a process name (except for the context level process) should be a verb phrase
    - usually three words: verb, modifier, noun
    - on a physical DFD, could be a complete sentence


1.0
Gather Data
Demographic Data
2.0
Compile Statistics
Survey Responses
3.0
Analyze Responses
Final Report

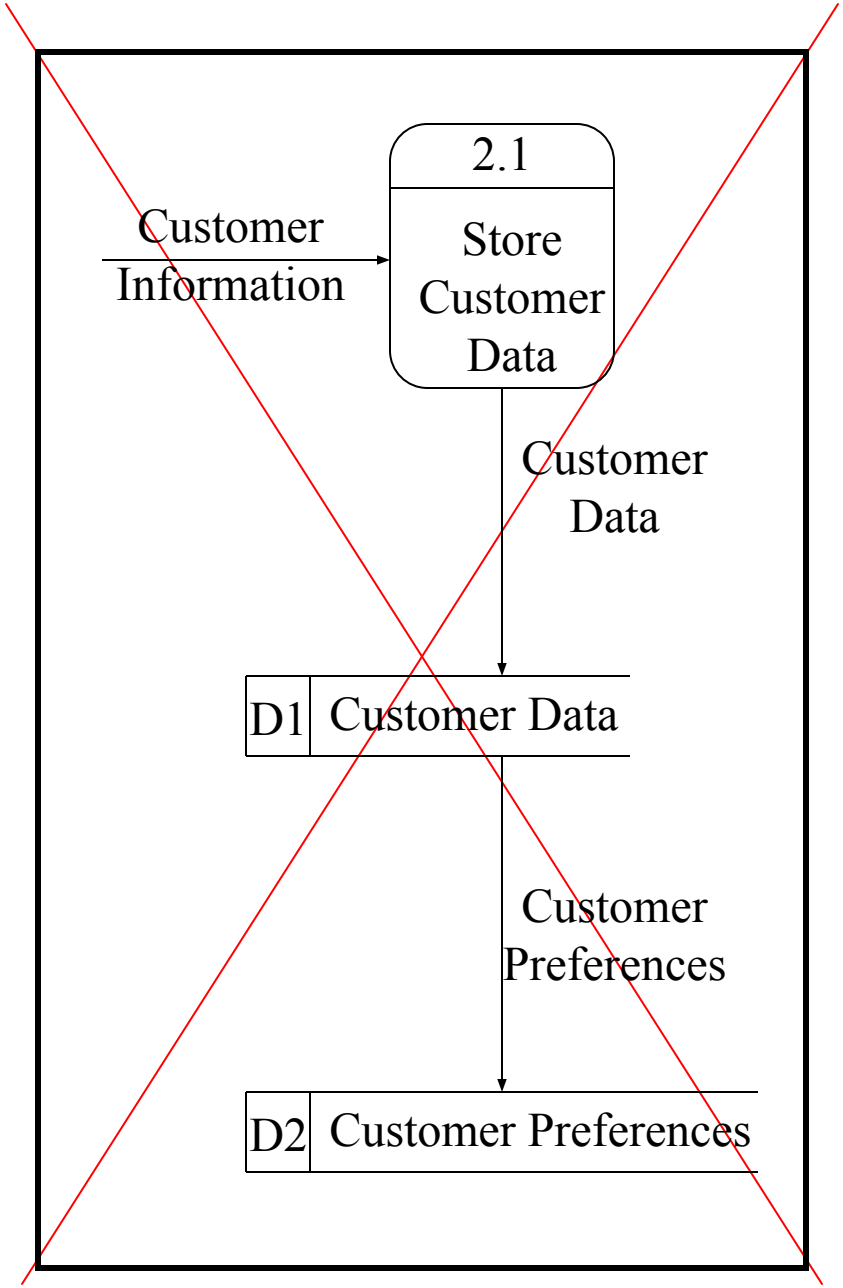

2.0
Visa Authorization
BETTER
2.0
Check Customer Credit
2.0
Total Records
BETTER
2.0
Total Sales Records
2.0
QA Process
BETTER
2.0
Inspect Finished Products

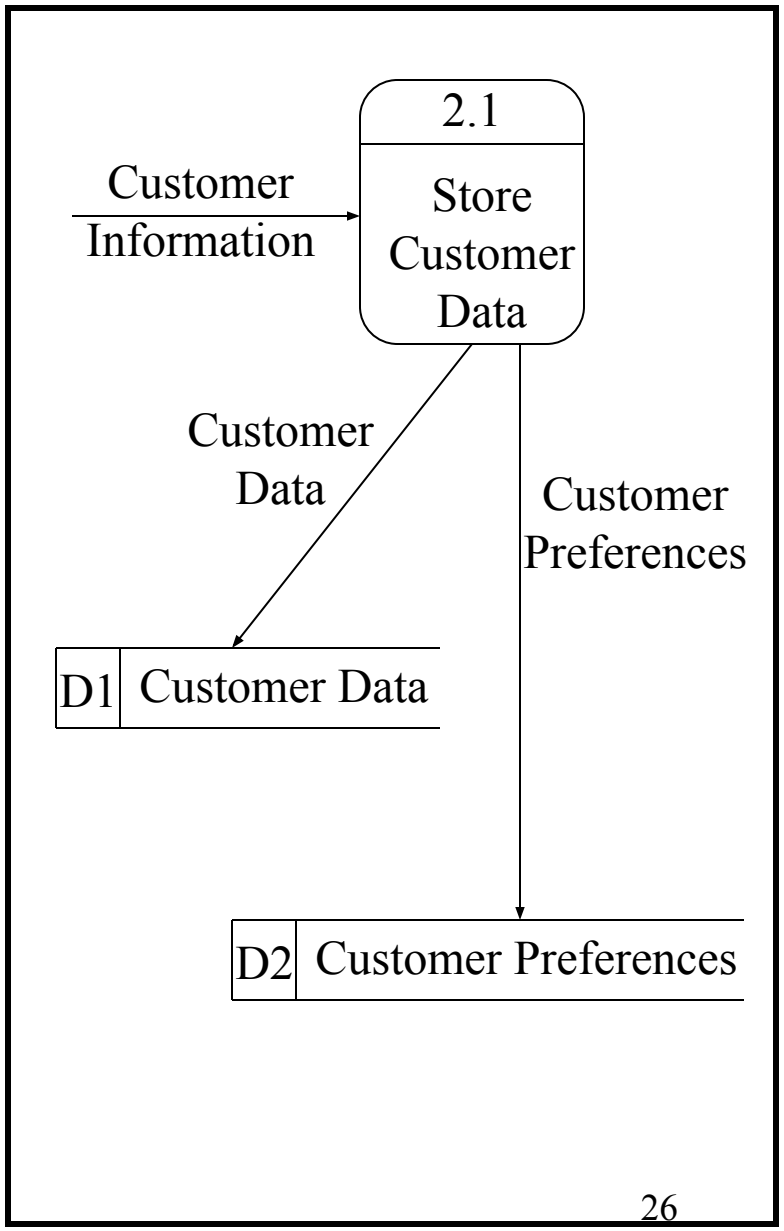
General DFD Rules

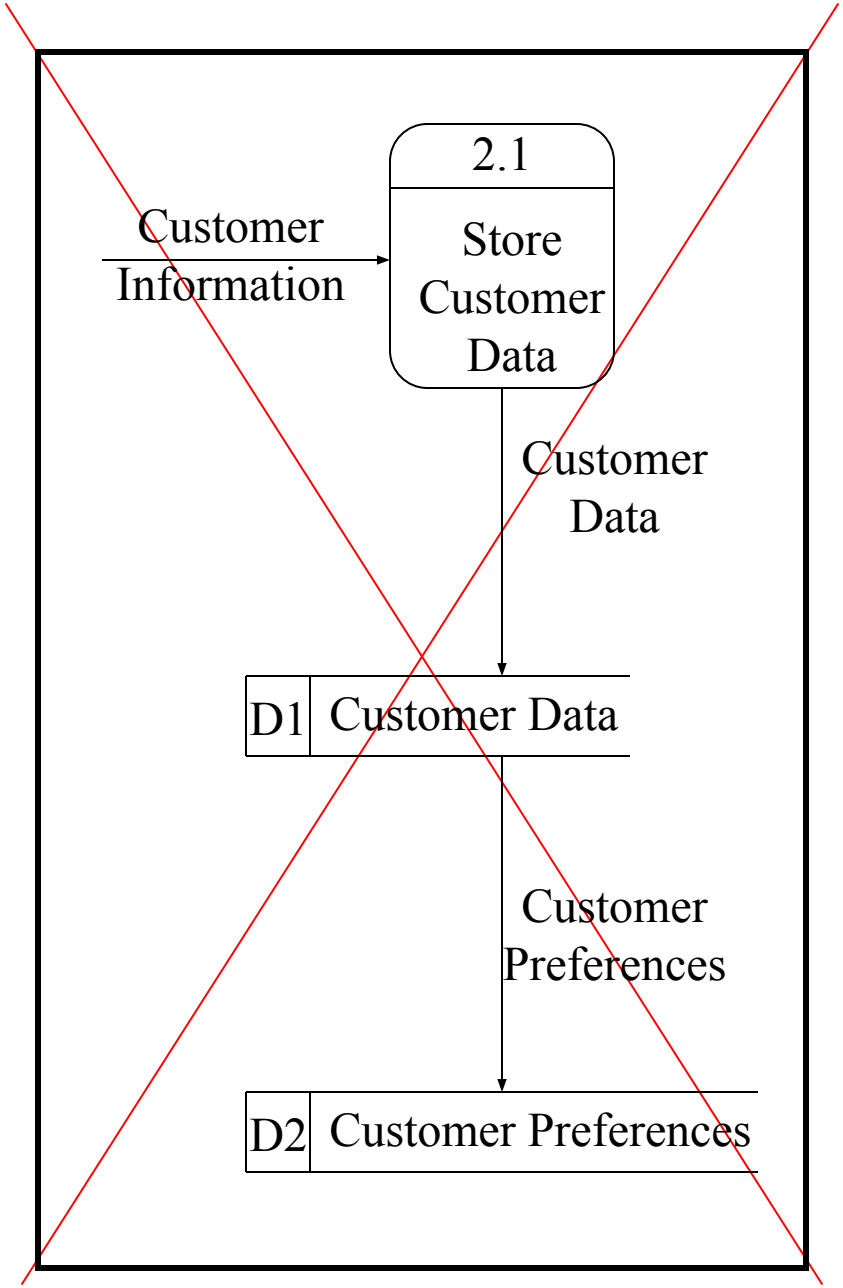
	YES	NO
A process to another process	✓	
A process to an external entity	✓	
A process to a data store	✓	
An external entity to another external entity		✓
An external entity to a data store		✓
A data store to another data store		✓

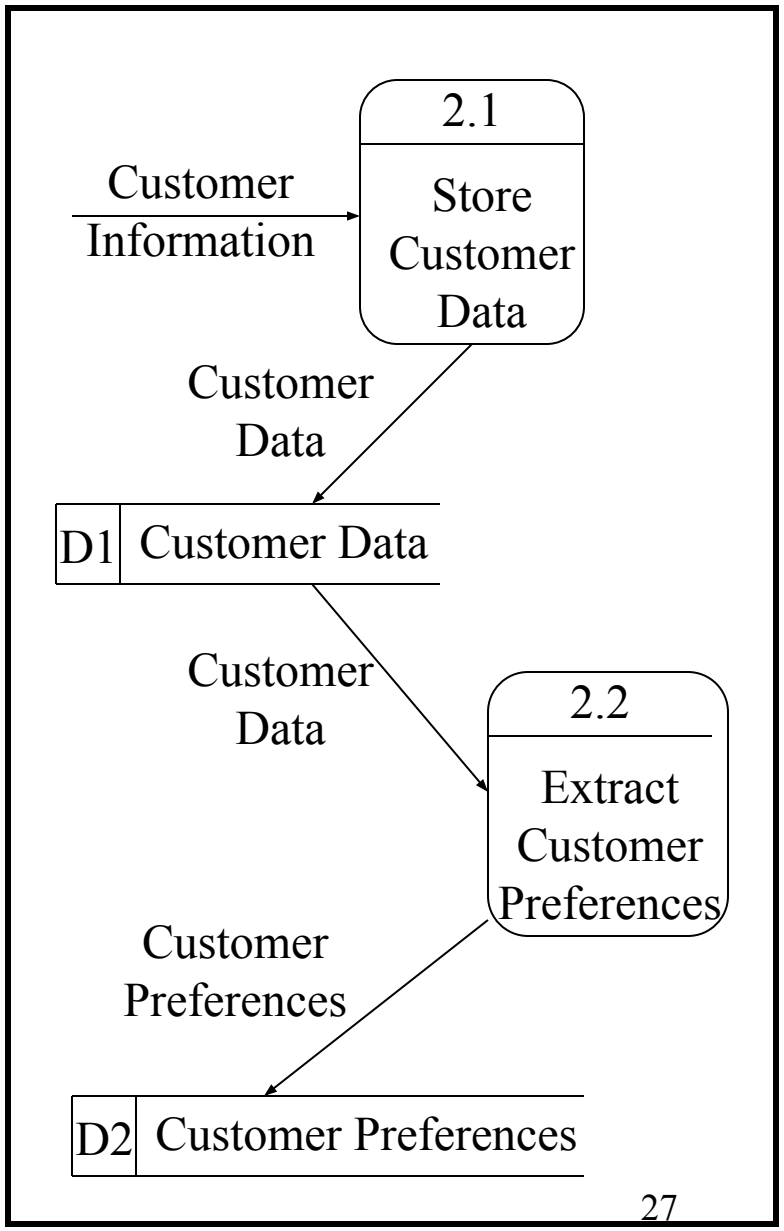
Data Flow Diagramming Rules

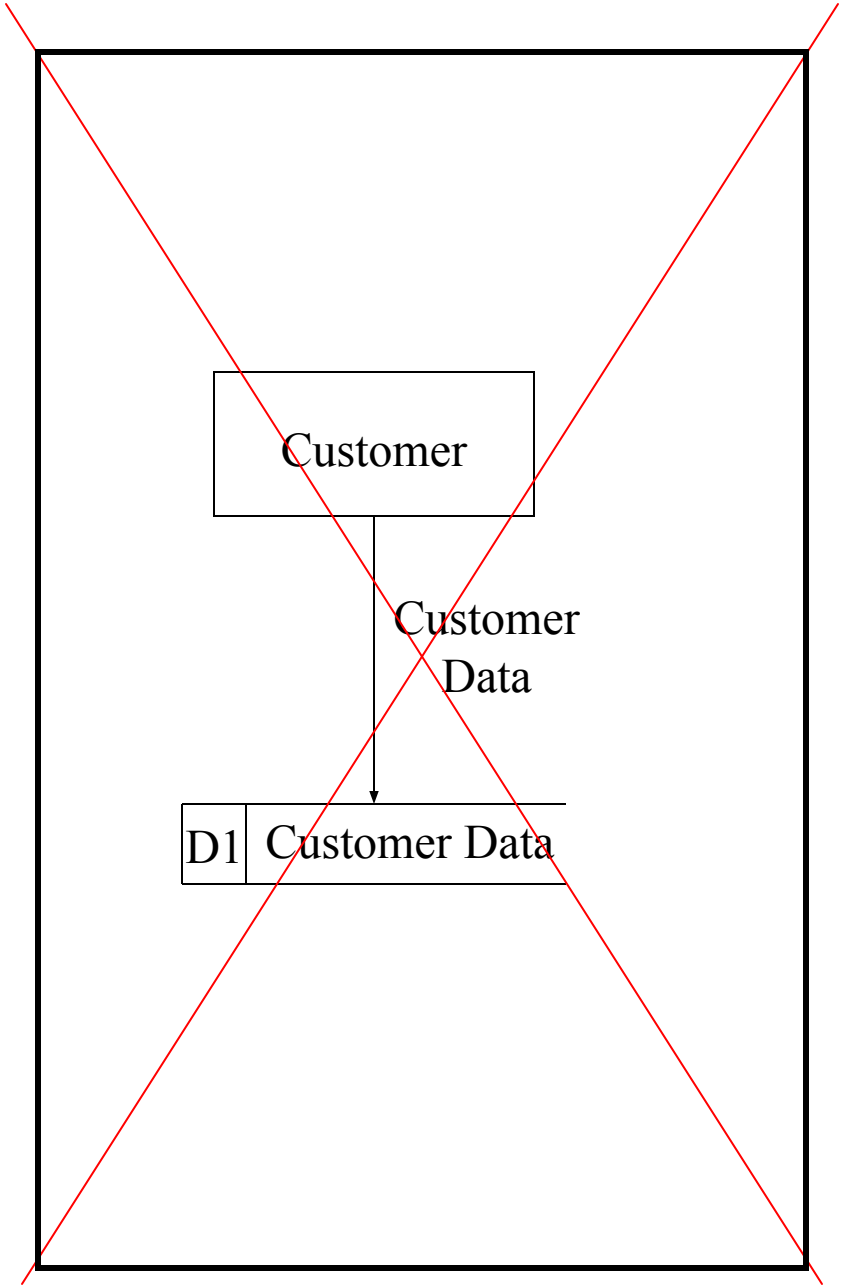
- Data stores and sources/sinks
  - no data flows between two data stores; must be a process in between
  - no data flows between a data store and a source or sink; must be a process in between
  - no data flows between two sources/sinks
    - such a data flow is not of interest, or
    - there is a process that moves that data

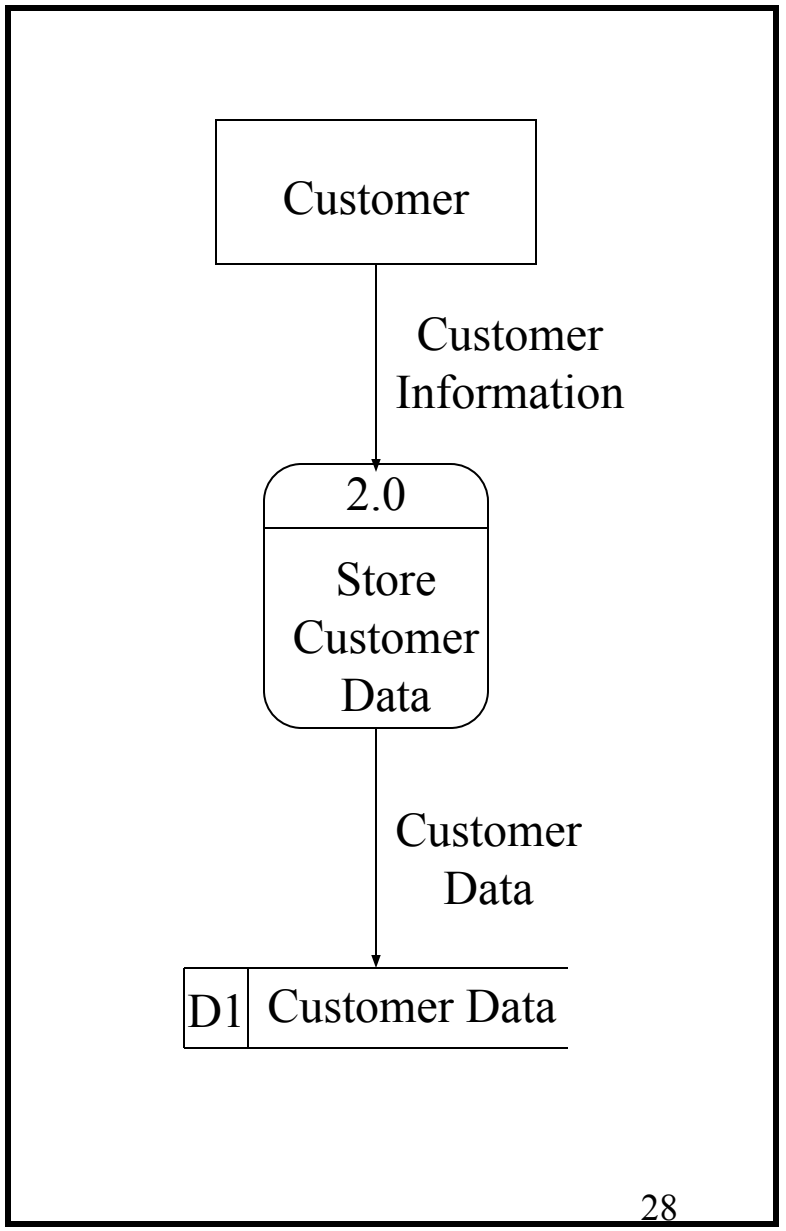

2.1
Store Customer Data
Customer Information
Customer Data
D1 Customer Data
Customer Preferences
D2 Customer Preferences

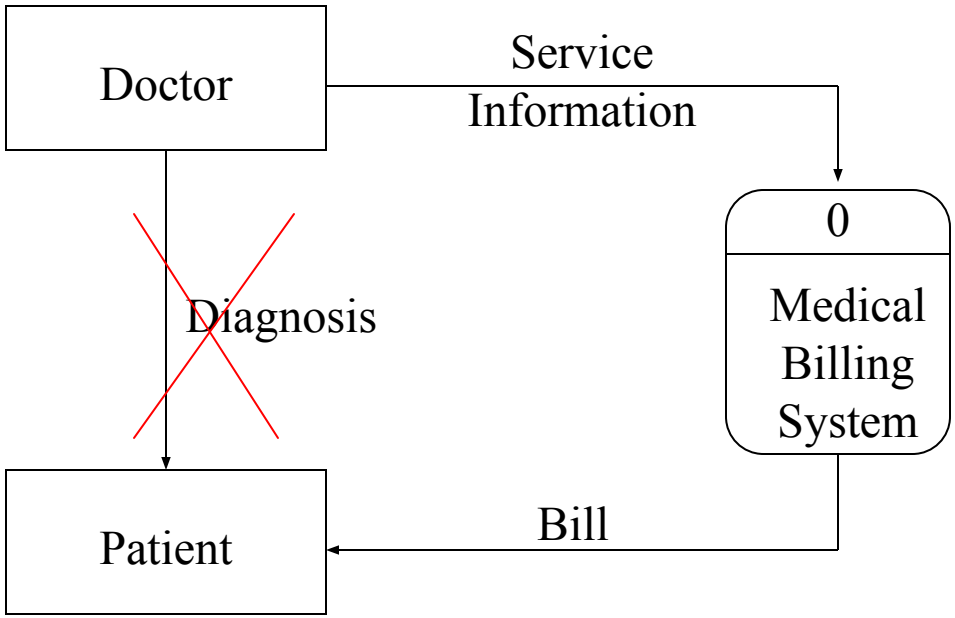

2.1
Store Customer Data
Customer Information
Customer Data
Customer Preferences
D1 Customer Data
D2 Customer Preferences


2.1
Store Customer Data
Customer Information
Customer Data
D1 Customer Data
Customer Preferences
D2 Customer Preferences


2.1
Store Customer Data
Customer Information
Customer Data
D1 Customer Data
Customer Data
2.2
Extract Customer Preferences
Customer Preferences
D2 Customer Preferences

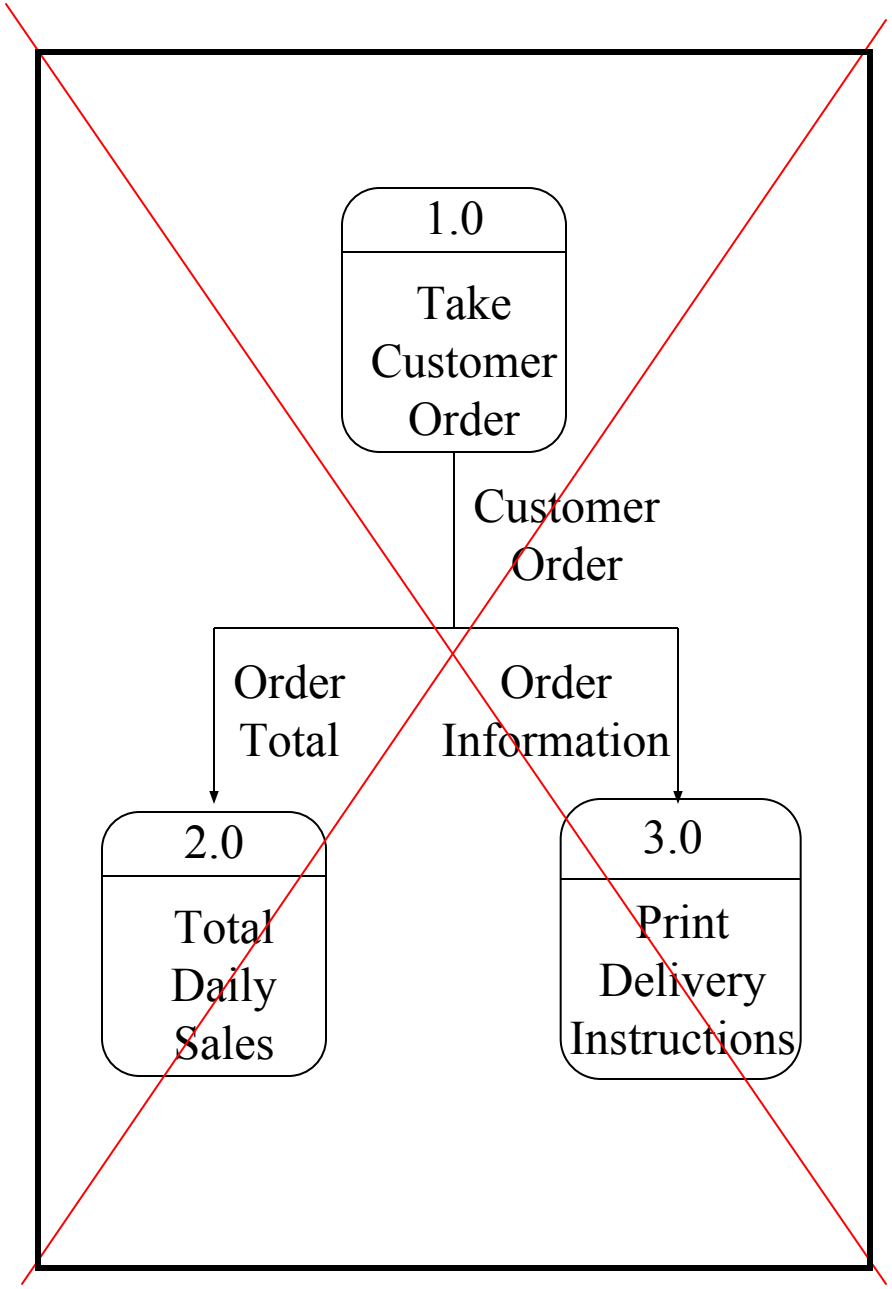

Customer
Customer Data
D1
Customer Data

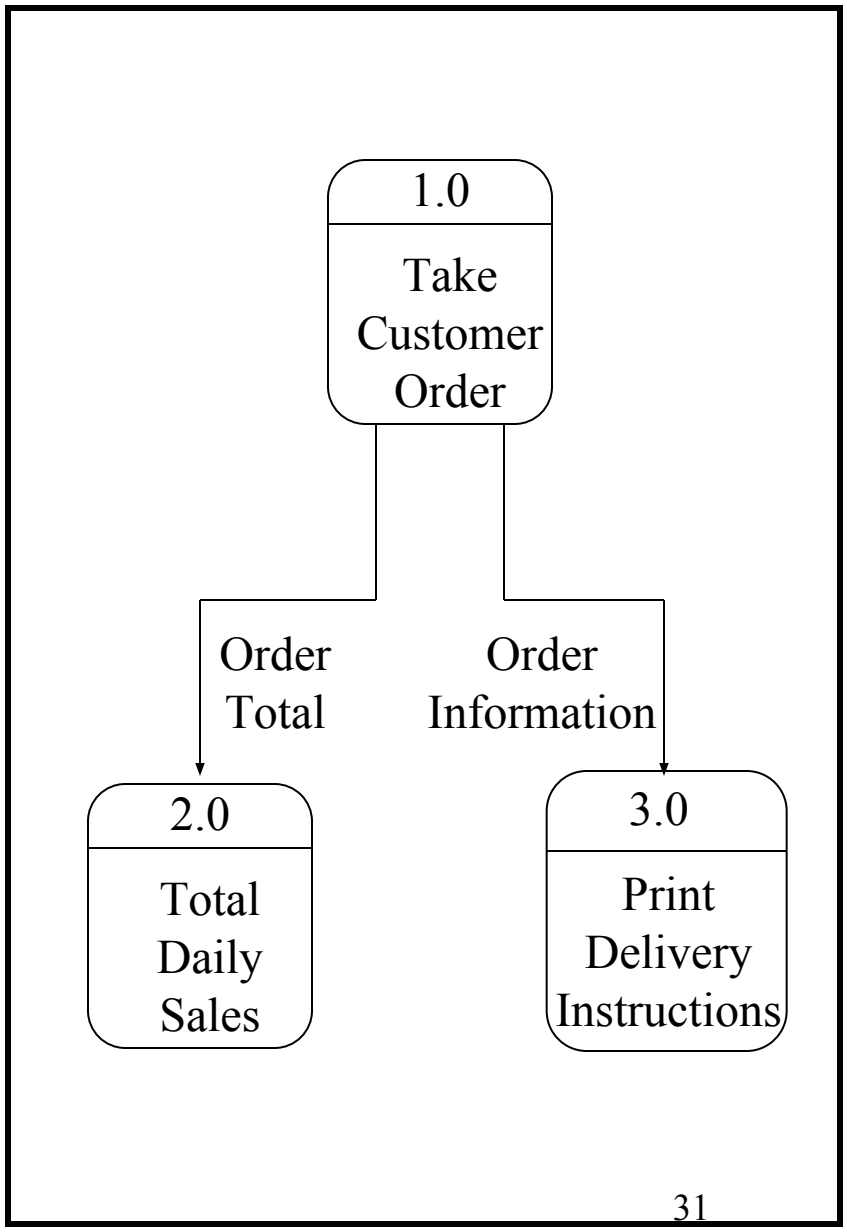

Customer
Customer Information
2.0
Store Customer Data
Customer Data
D1
Customer Data

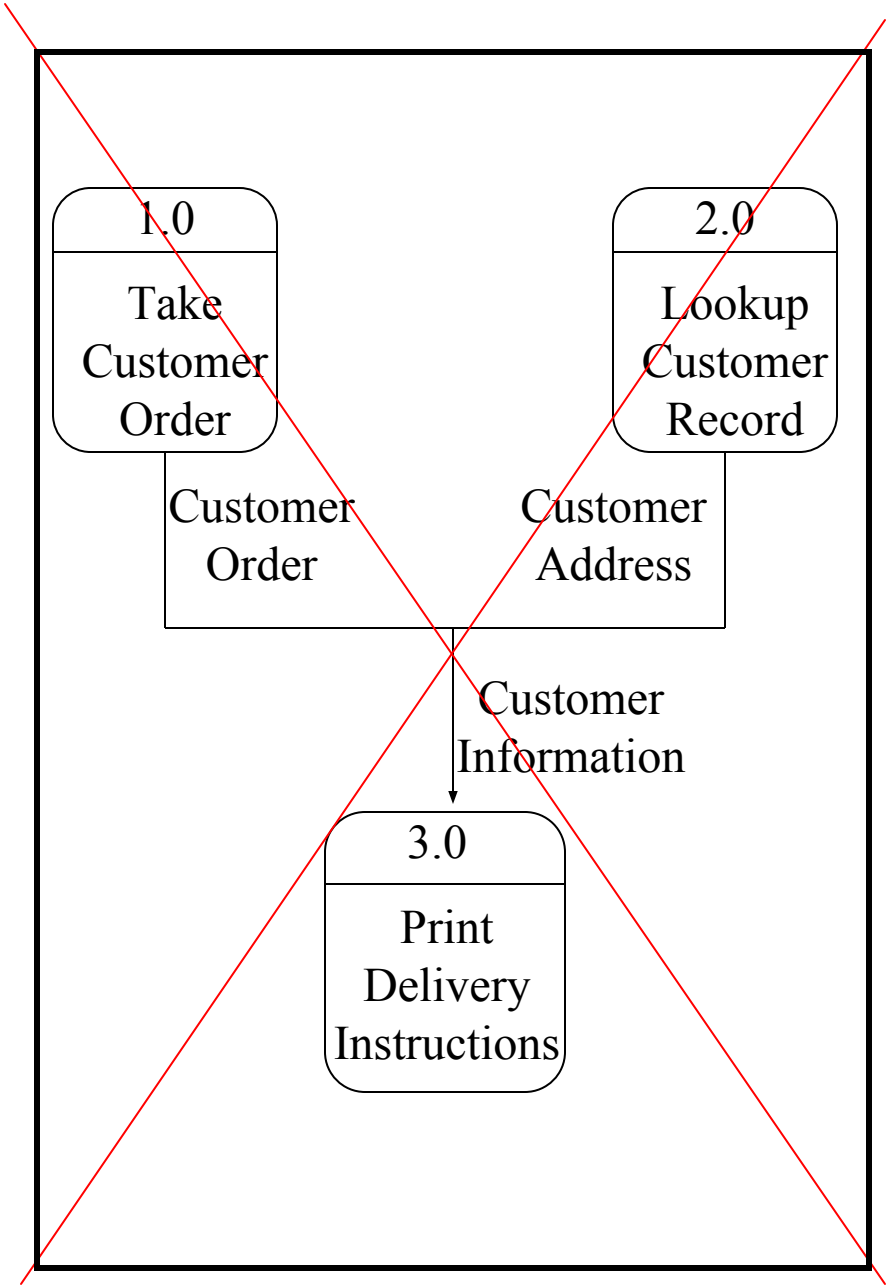

Doctor
Service
Information
0
Medical
Billing
System
Diagnosis
Patient
Bill

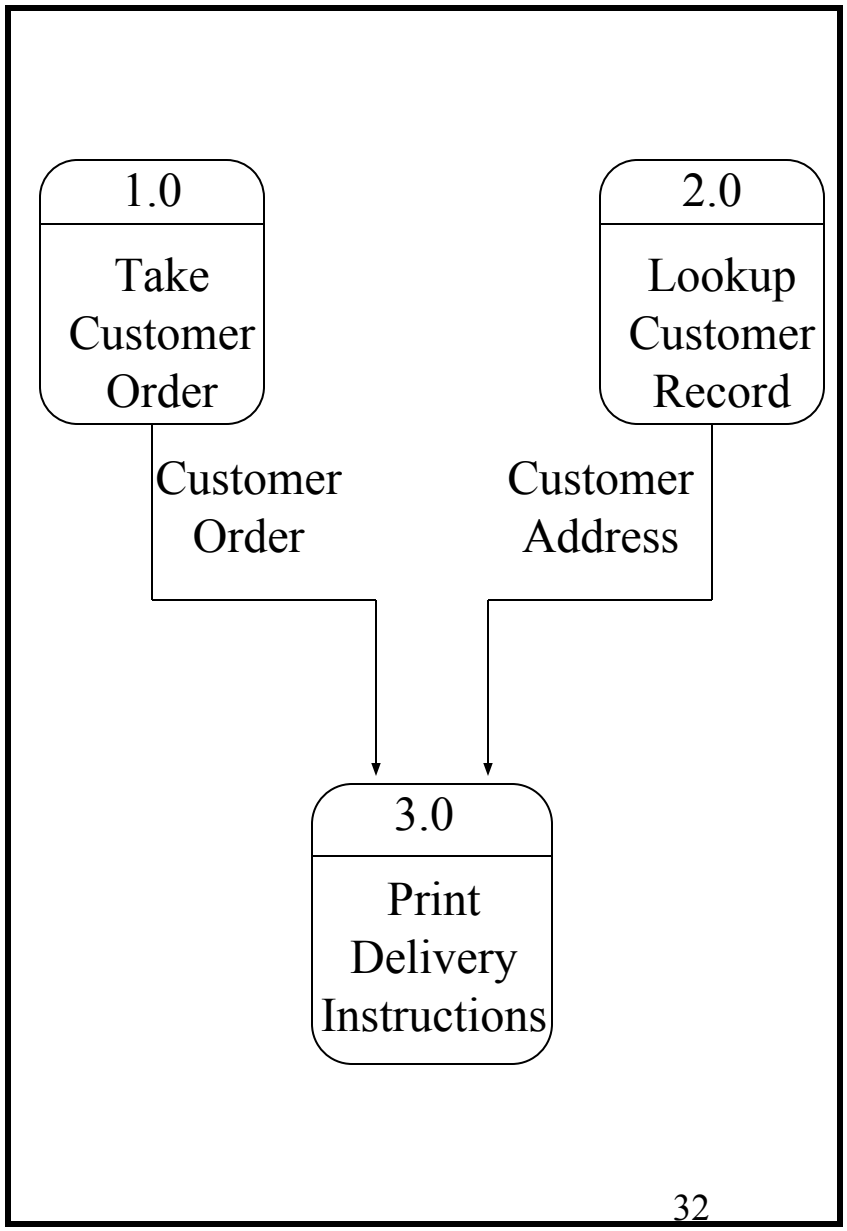
Data Flow Diagramming Rules

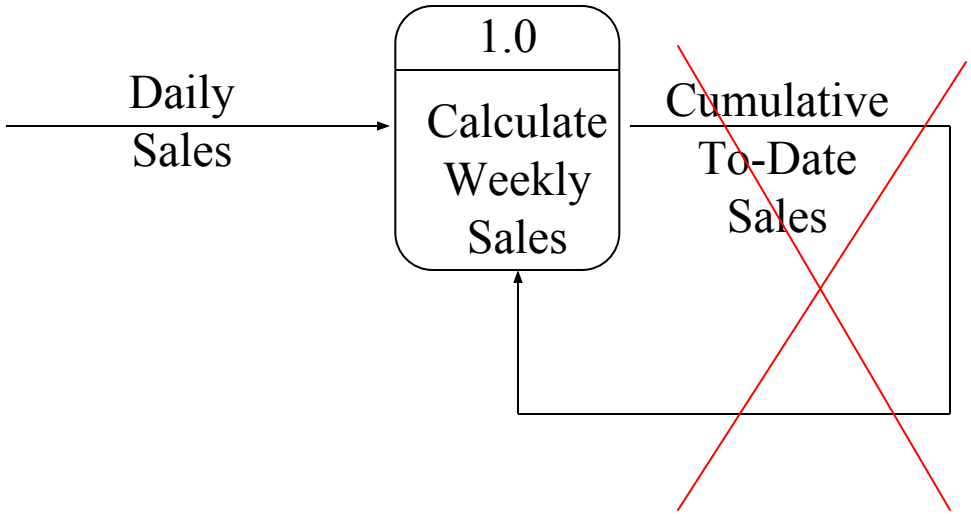
- Data flows
  - data flows are unidirectional
  - a data flow may fork, delivering exactly the same data to two different destinations
  - two data flows may join to form one only if the original two are exactly the same
  - no recursive data flows
  - data flows (and data stores and sources/sinks) are labelled with noun phrases


1.0
Take Customer Order
Customer Order
Order Total
Order Information
2.0
Total Daily Sales
3.0
Print Delivery Instructions


1.0
Take Customer Order
Order Total
Order Information
2.0
Total Daily Sales
3.0
Print Delivery Instructions

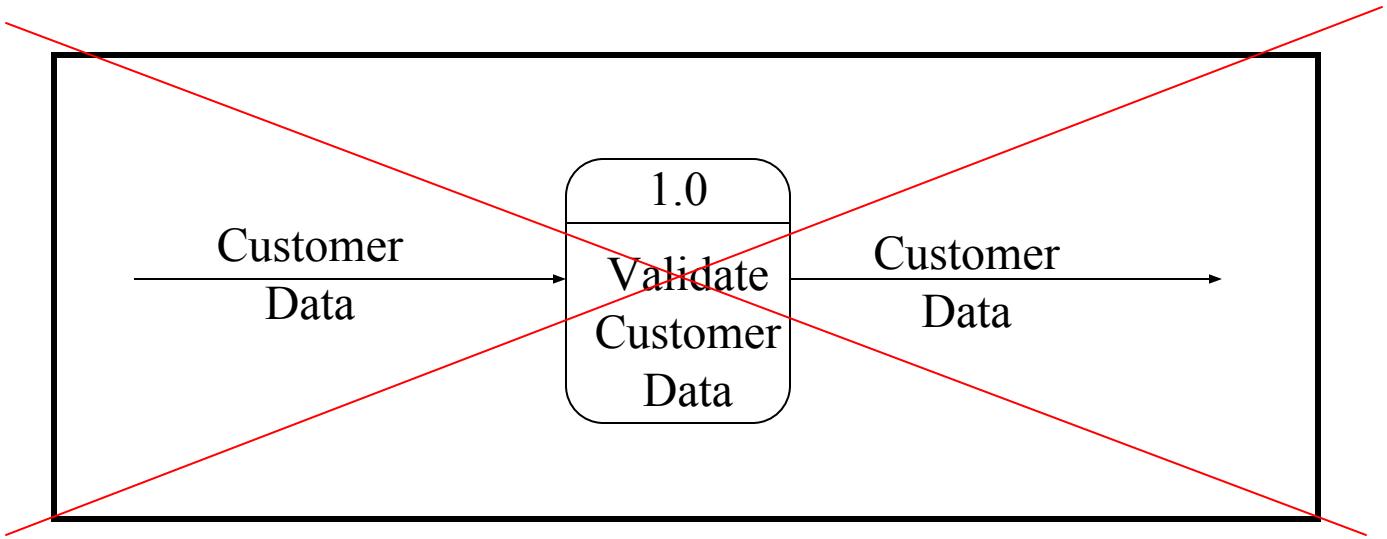

1.0
Take Customer Order
2.0
Lookup Customer Record
Customer Order
Customer Address
Customer Information
3.0
Print Delivery Instructions

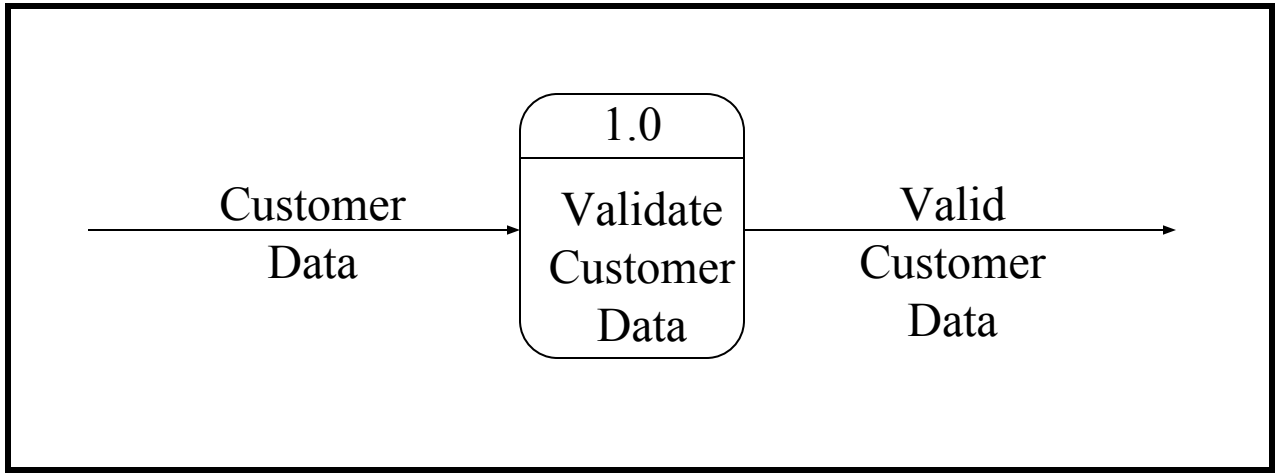

1.0
Take Customer Order
2.0
Lookup Customer Record
Customer Order
Customer Address
3.0
Print Delivery Instructions

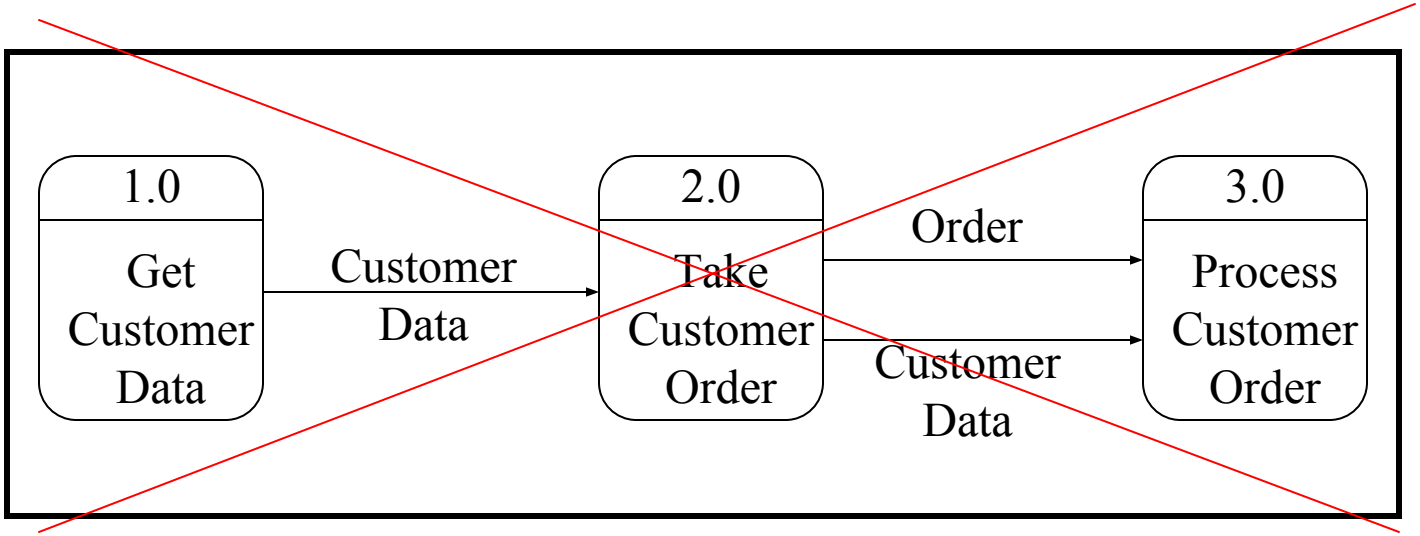

1.0
Daily
Sales
Calculate
Weekly
Sales
Cumulative
To-Date
Sales

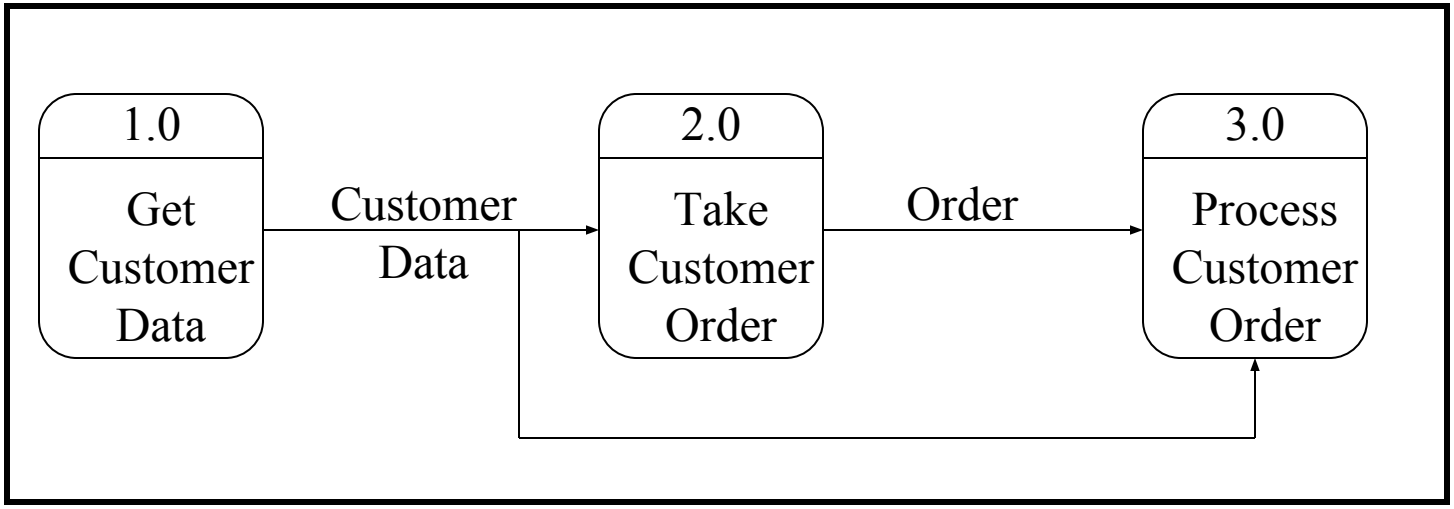
Data Flow Diagramming Guidelines

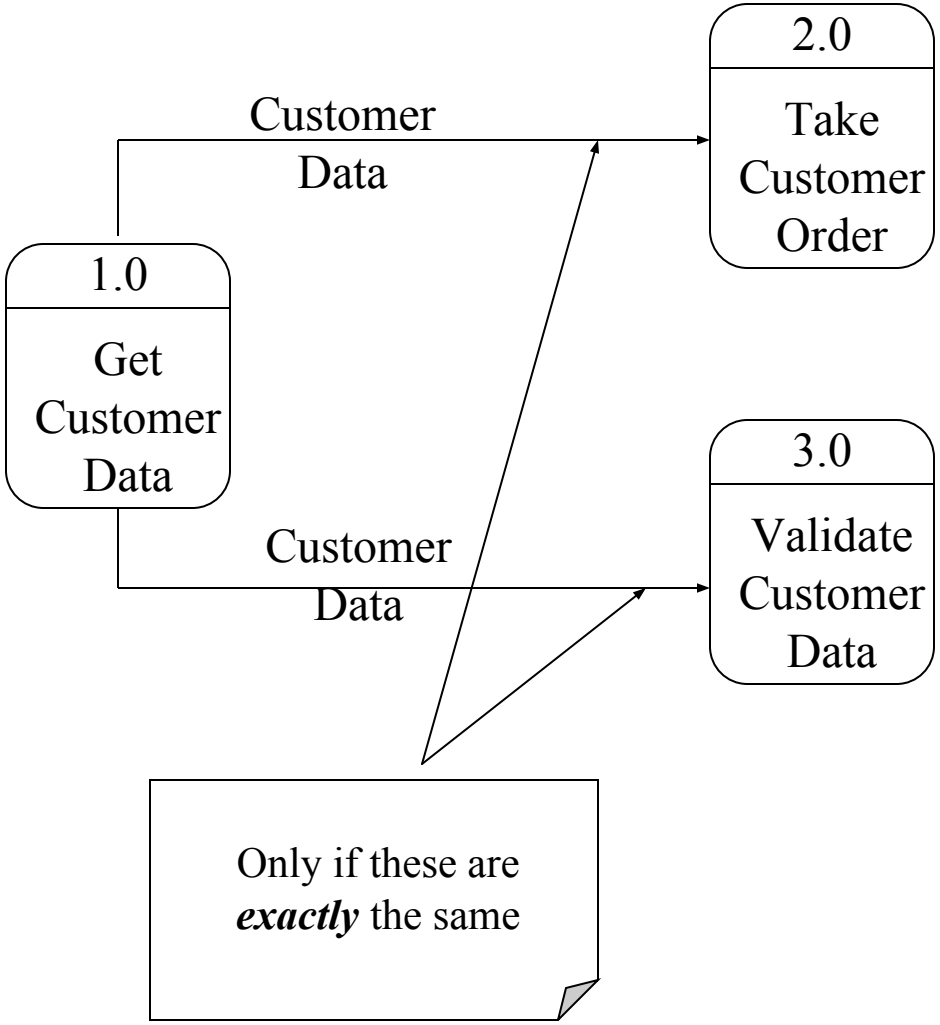
- The inputs to a process are different from the outputs
- Every object in a DFD has a unique name


1.0
Customer
Data
Validate
Customer
Data
Customer
Data


1.0
Customer
Data
Validate
Customer
Data
Valid
Customer
Data

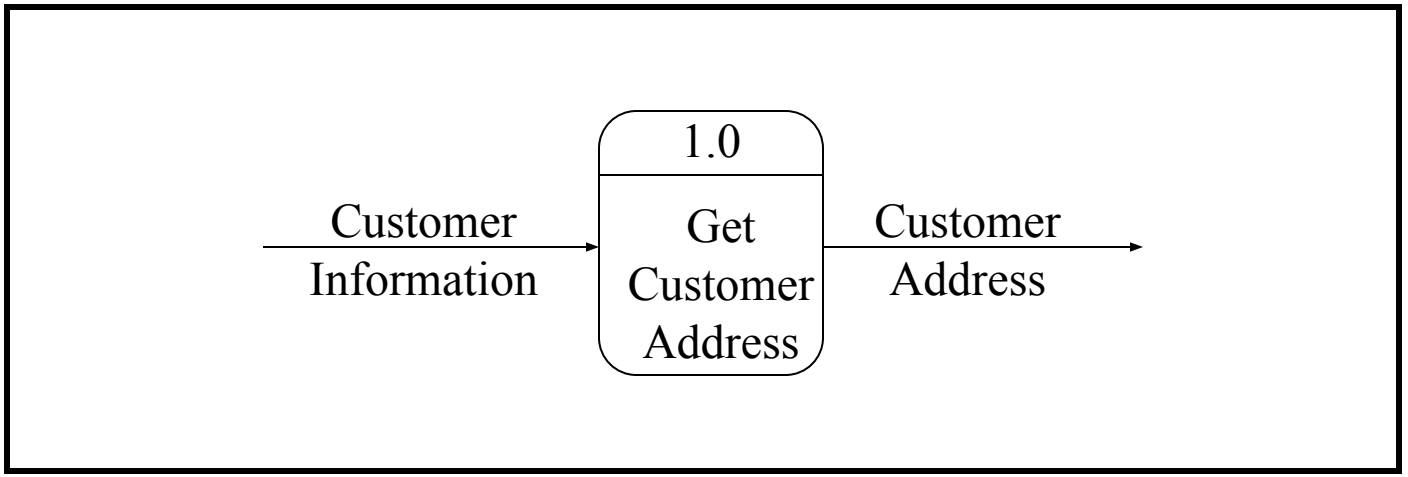

1.0
Get Customer Data
Customer Data
2.0
Take Customer Order
Order
Customer Data
3.0
Process Customer Order

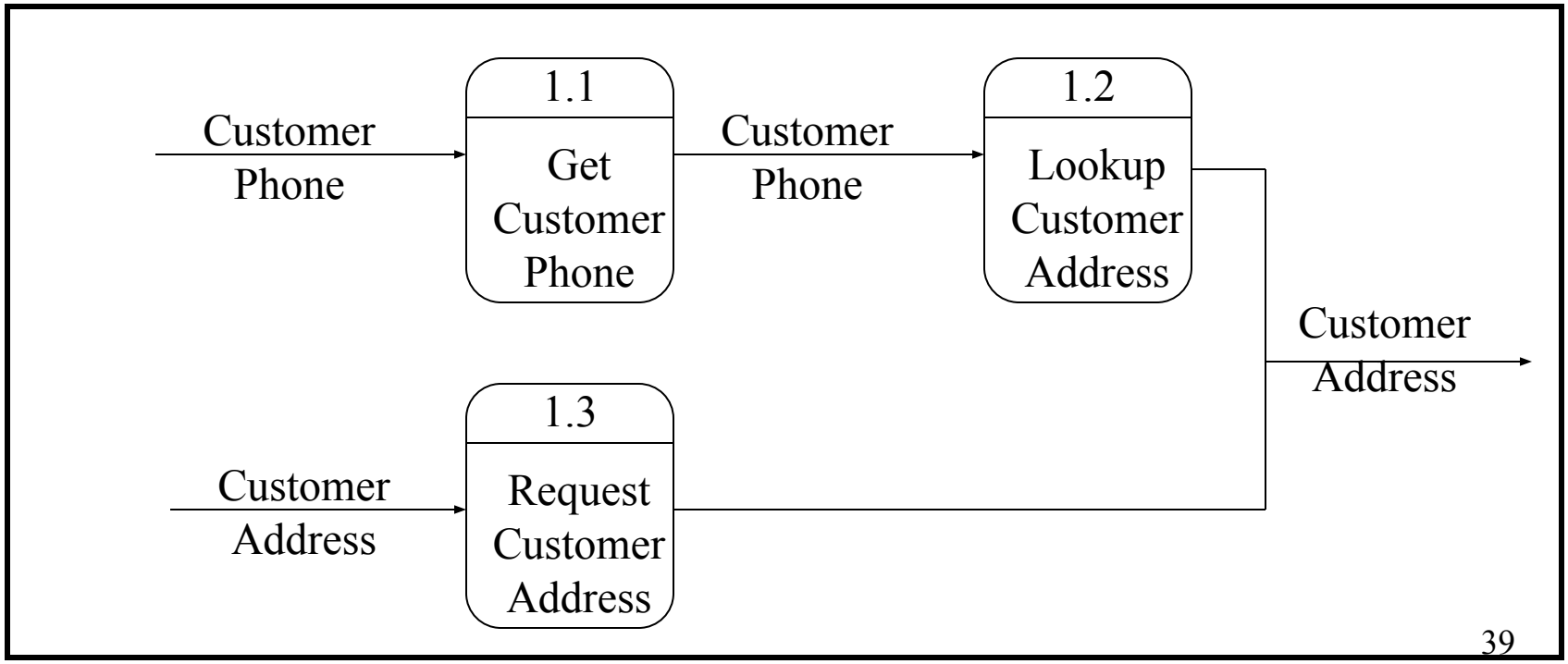

1.0
Get Customer Data
Customer Data
2.0
Take Customer Order
Order
3.0
Process Customer Order

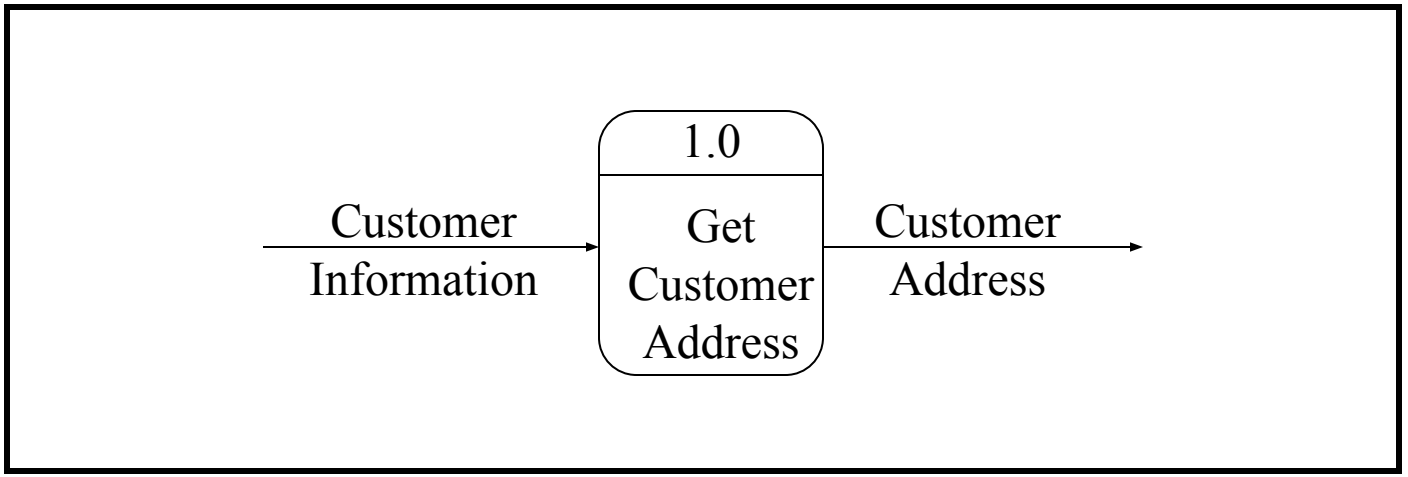

2.0
Take Customer Order
Customer Data
1.0
Get Customer Data
3.0
Validate Customer Data
Customer Data
Only if these are ***exactly*** the same

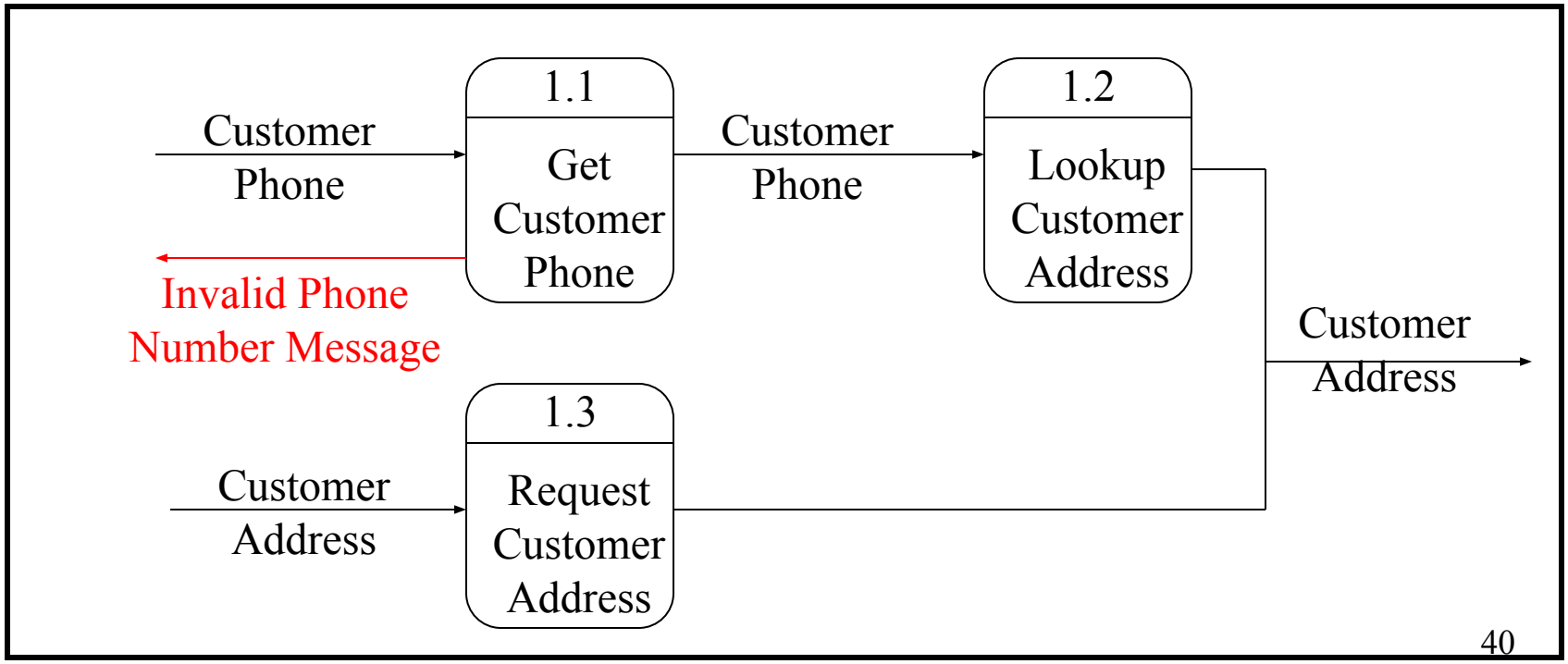
Data Flow Diagramming Guidelines

- A data flow at one level may be decomposed at a lower level
- All data coming into and out of a process must be accounted for
- On low-level DFDs, new data flows can be added to represent exceptional situations


1.0
Customer Information
Get Customer Address
Customer Address


1.1
Customer Phone
Get Customer Phone
Customer Phone
1.2
Lookup Customer Address
Customer Address
1.3
Customer Address
Request Customer Address


1.0
Customer Information
Get Customer Address
Customer Address


1.1
Customer Phone
Get Customer Phone
Invalid Phone Number Message
1.2
Customer Phone
Lookup Customer Address
Customer Address
1.3
Customer Address
Request Customer Address

Data Elements

- Indivisible pieces of data
- Data flows and data stores are made up of data elements
- Like attributes on an ER diagram
- The data elements of a data flow flowing in or out of a data store must be a subset of the data elements in that data store

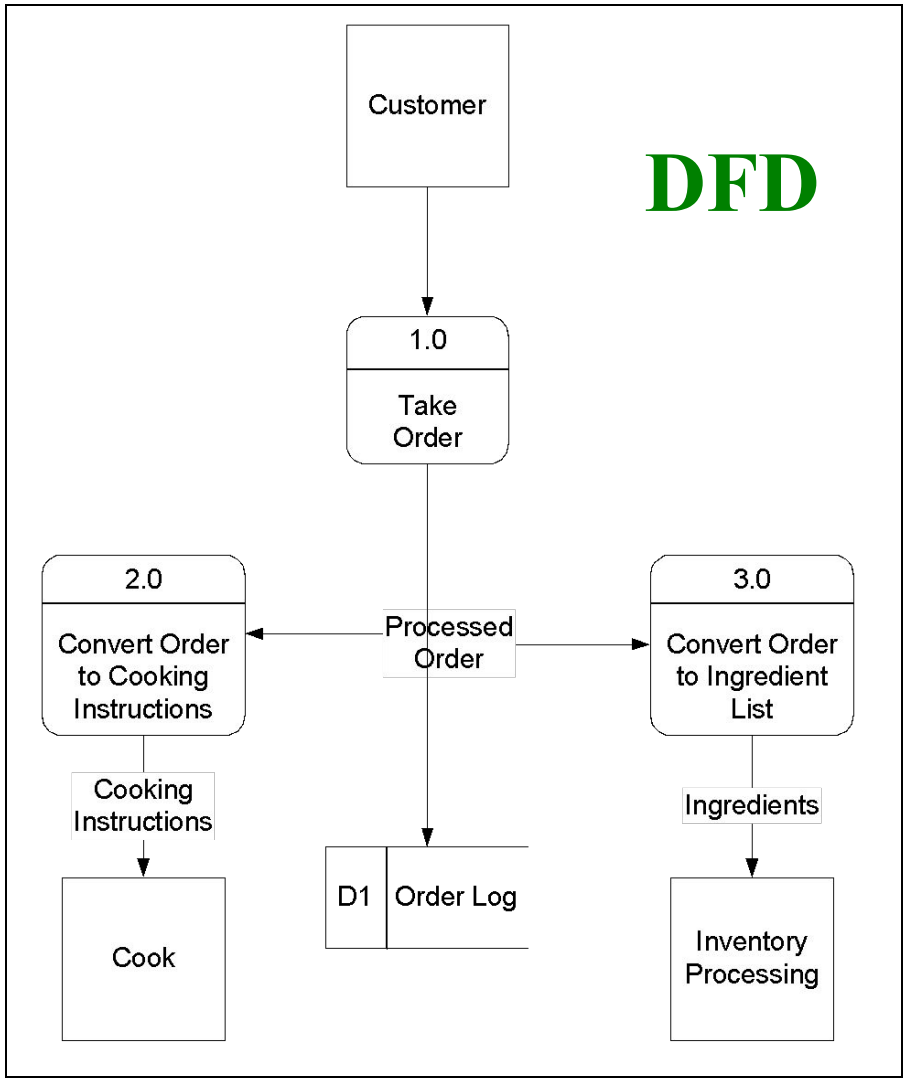
DFDs and ERDs

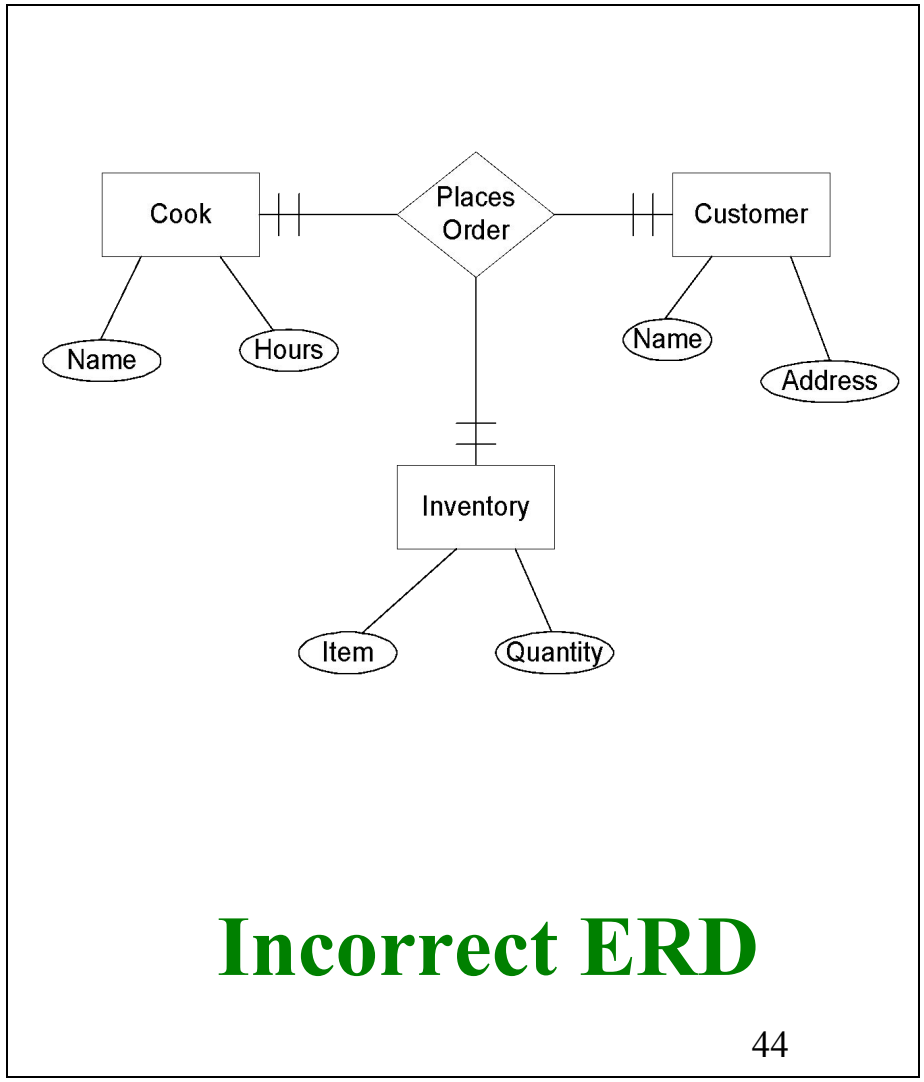
- DFDs and ERDs are both used to model systems, but they show two very different perspectives on the system
- A DFD shows what the system ***does*** as well as the ***data*** that the system manipulates
- An ERD shows **only** the ***data*** that the system manipulates.

DFDs and ERDs (cont.)

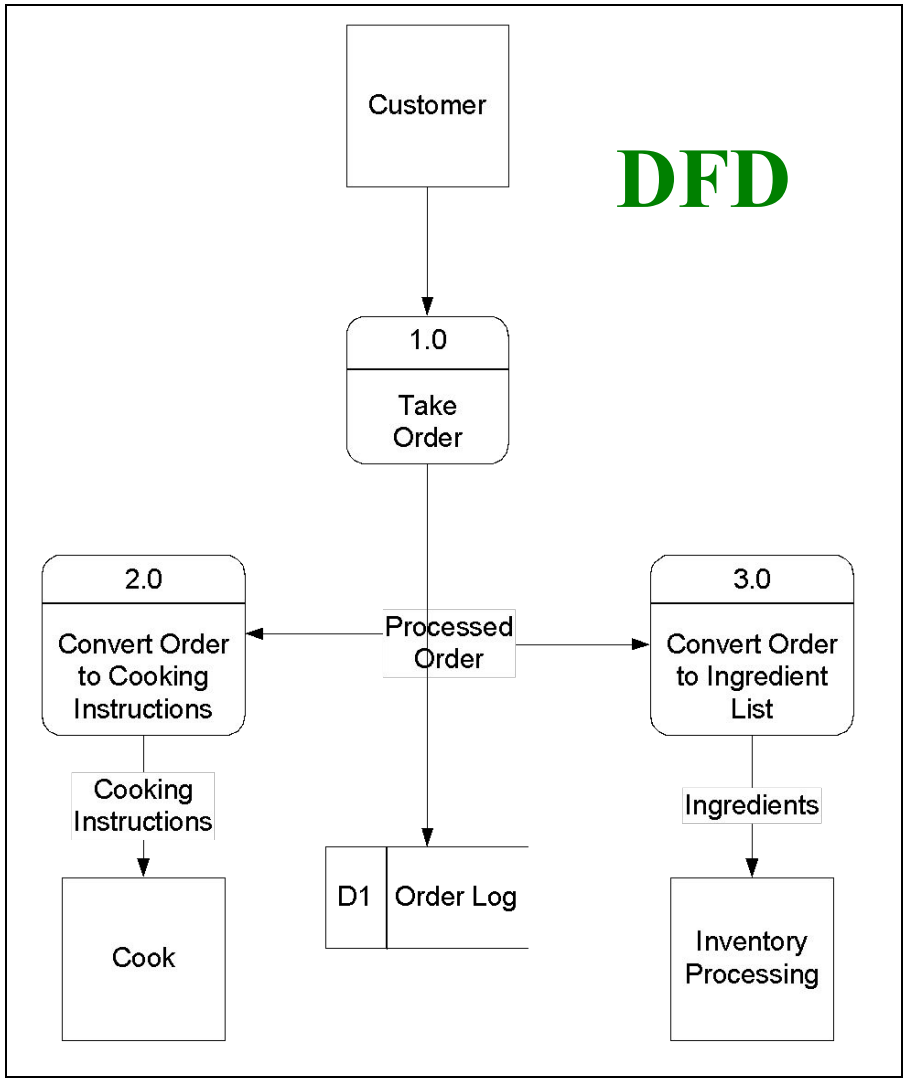
- Entities on an ERD often (but not always) correspond to data stores on a DFD
- Attributes on an ERD usually correspond to data elements (listed in the data dictionary) that make up the data store and data flows on a DFD
- Relationships on an ERD **do not** correspond to processes on a DFD.
- Sources and sinks on a DFD usually **do not** show up as entities on an ERD

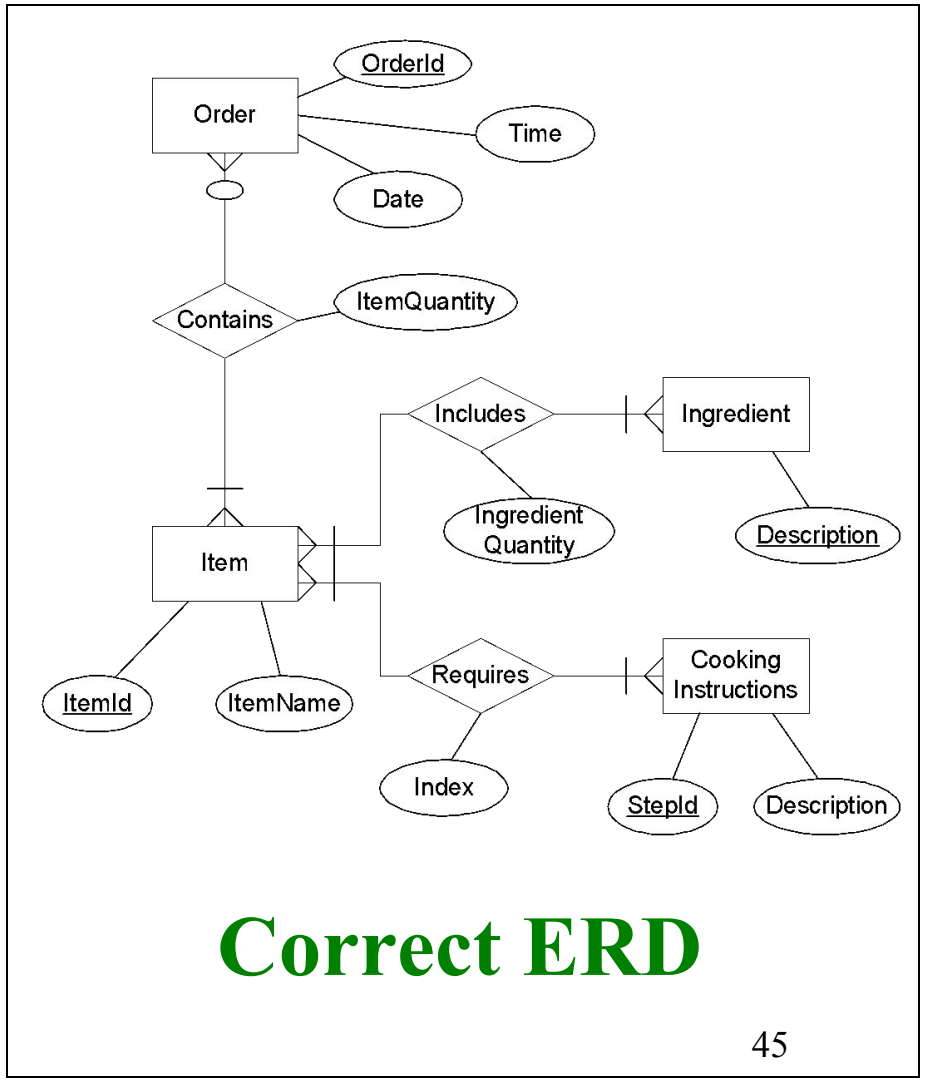
Example DFD and ERD





Example DFD and ERD



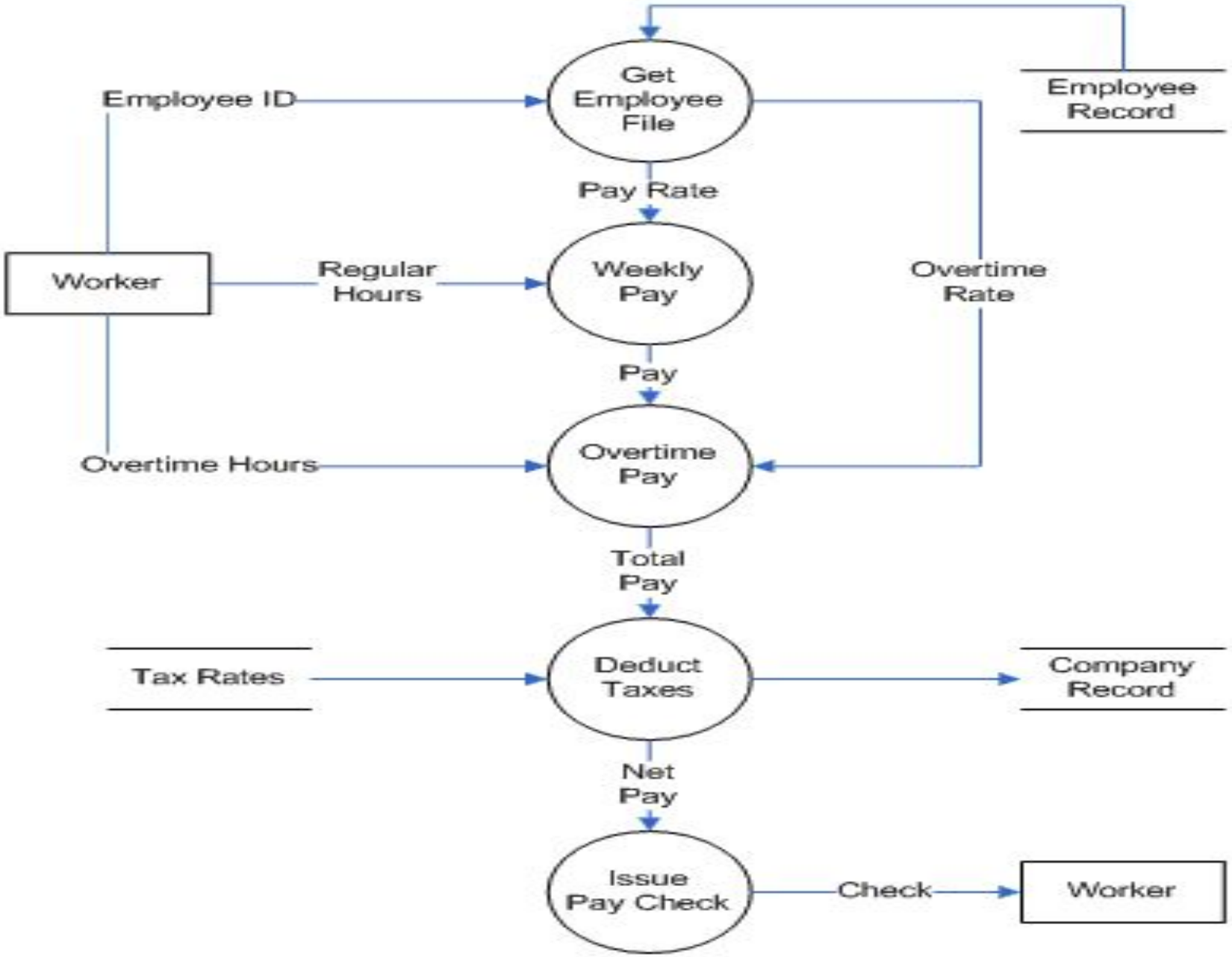


Advantages of DFDs

- Simple graphical techniques which are easy to understand
- Helps define the boundaries of the system
- Useful for communicating current system knowledge to users
- Explains the logic behind the data flow within the system
- Used as the part of system documentation file

Example 1

- In the following **DFD** there is one basic input data flow, the weekly timesheet, which originates from the source worker. The basic output is the paycheck, the sink for which is also the worker. In this system, first the employee's record is retrieved, using the employee ID, which is contained in the timesheet. From the employee record, the rate of payment and overtime are obtained. These rates and the regular and overtime hours (from the timesheet) are used to compute the pay. After the total pay is determined, taxes are deducted. To compute the tax deduction, information from the tax-rate file is used. The amount of tax deducted is recorded in the employee and company records. Finally, the paycheck is issued for the net pay. The amount paid is also recorded in company records.


Get Employee File
Employee Record
Employee ID
Pay Rate
Worker
Regular Hours
Weekly Pay
Overtime Rate
Pay
Overtime Hours
Overtime Pay
Total Pay
Tax Rates
Deduct Taxes
Company Record
Net Pay
Issue Pay Check
Check
Worker