Data Flow Diagrams

- **Data flow diagram:** is a graphical modeling tool to depict the flow of data through a system and the work or processing performed by that system.
  - What's the system doing?
- Use a limited number of symbols.
- Do not depict management or operational elements of a system.

Data Flow Diagram (DFD)

- Key points for modeling
  - **How data moves** through the organization
  - **Relationships** between various **data flows**
  - **Storage of data**

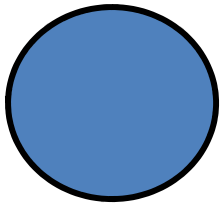
- There are **no FIXED rules** about how a DFD should be developed…

Notations of DFD

DE MARCO & YOURDON NOTATIONS

External Entity

Data Flow



Process

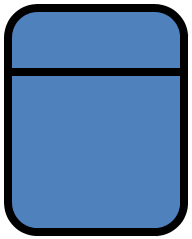
Data Store

Notations of DFD

GANE & SARSON NOTATIONS (more universal)

External Entity

Data Flow



Process

Data Store

Process Logic

- DFDs are effective tools for identifying processes, but are not good at showing all the detail logic inside those processes.
  - Not always easy to know what the lowest logical level is (see next slide)
- However, it is better than:
  - Flowcharts and Pseudocode (difficult for users to understand).

The Levels

Levels	Description	Explanation
Level 0	Context diagram	Contains only one process
Level 1	Overview diagram	Utilizes all four elements
Level 2	Detailed diagram	A breakdown of a level 2 process

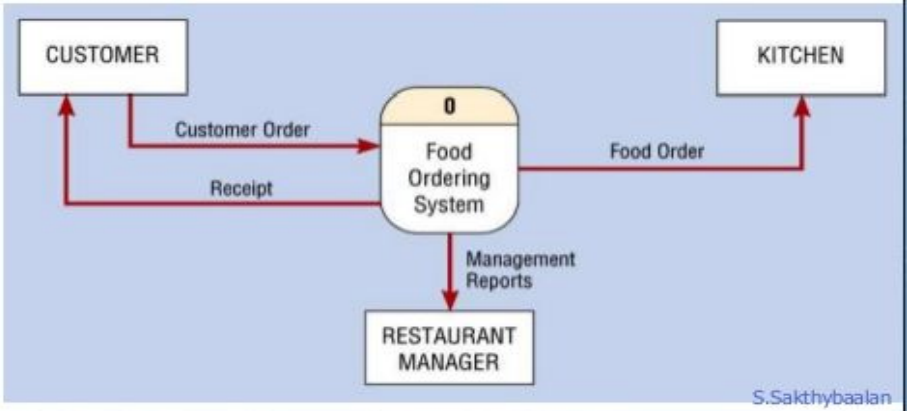
There is no rule as to how many levels of DFD that can be used.

A Context Diagram (Level 0)

- The major information flows between the entities and the system.
- A Context Diagram addresses only one process.

Rules for Level 0 Diagram :

- 1 process represents the entire system.
- Data arrows show input and output.
- Data Stores NOT shown. They are within the system.



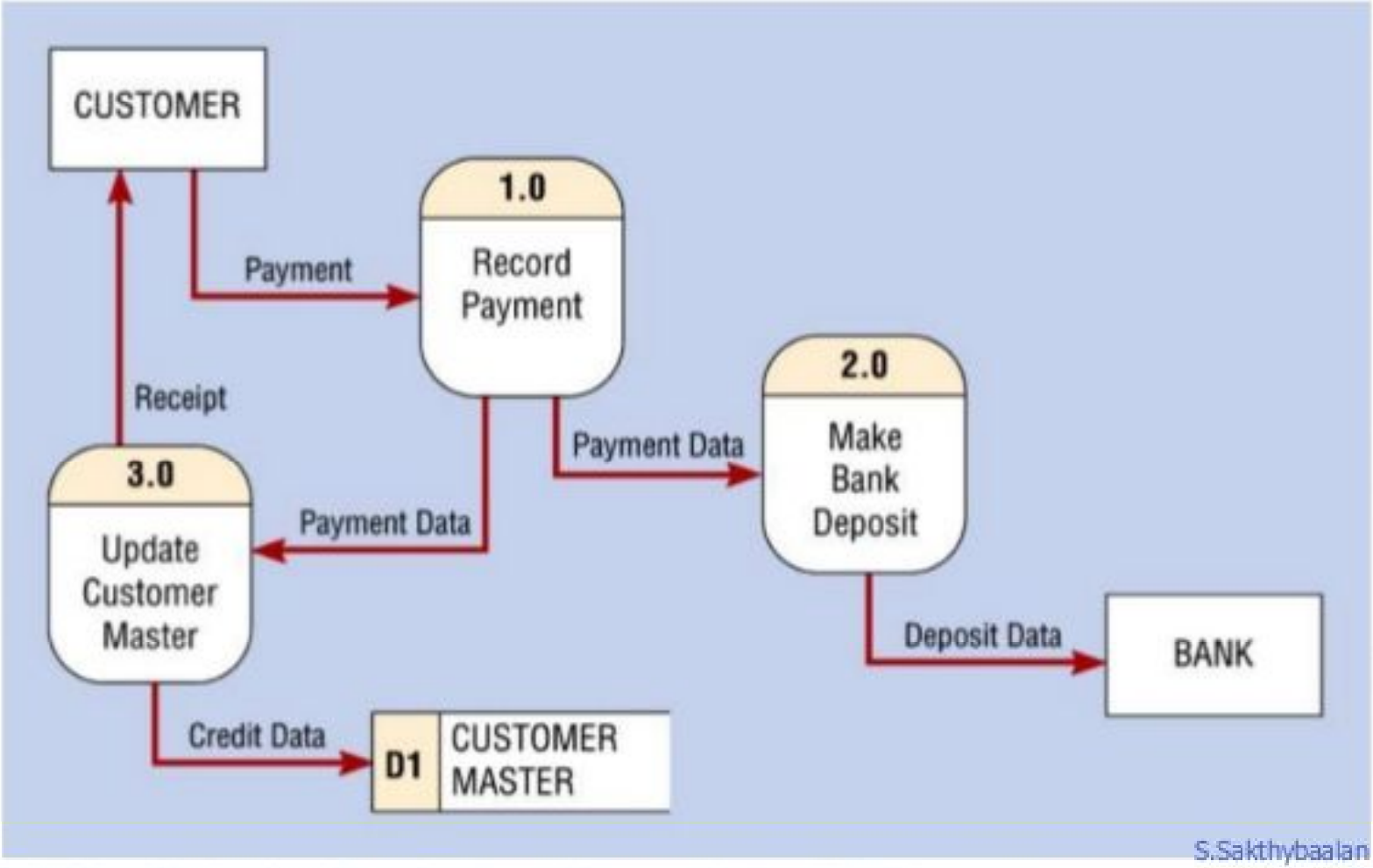
Rules for Level 1 Diagram :

- Level 1 DFD, must balance with the context diagram it describes.
- Input going into a process are different from outputs leaving the process.
- Data stores are first shown at this level.

Rules for Level 2 Diagram :

- Level 2 DFD must balance with the Level 1 it describes.
- Input going into a process are different from outputs leaving the process.
- Continue to show data stores.

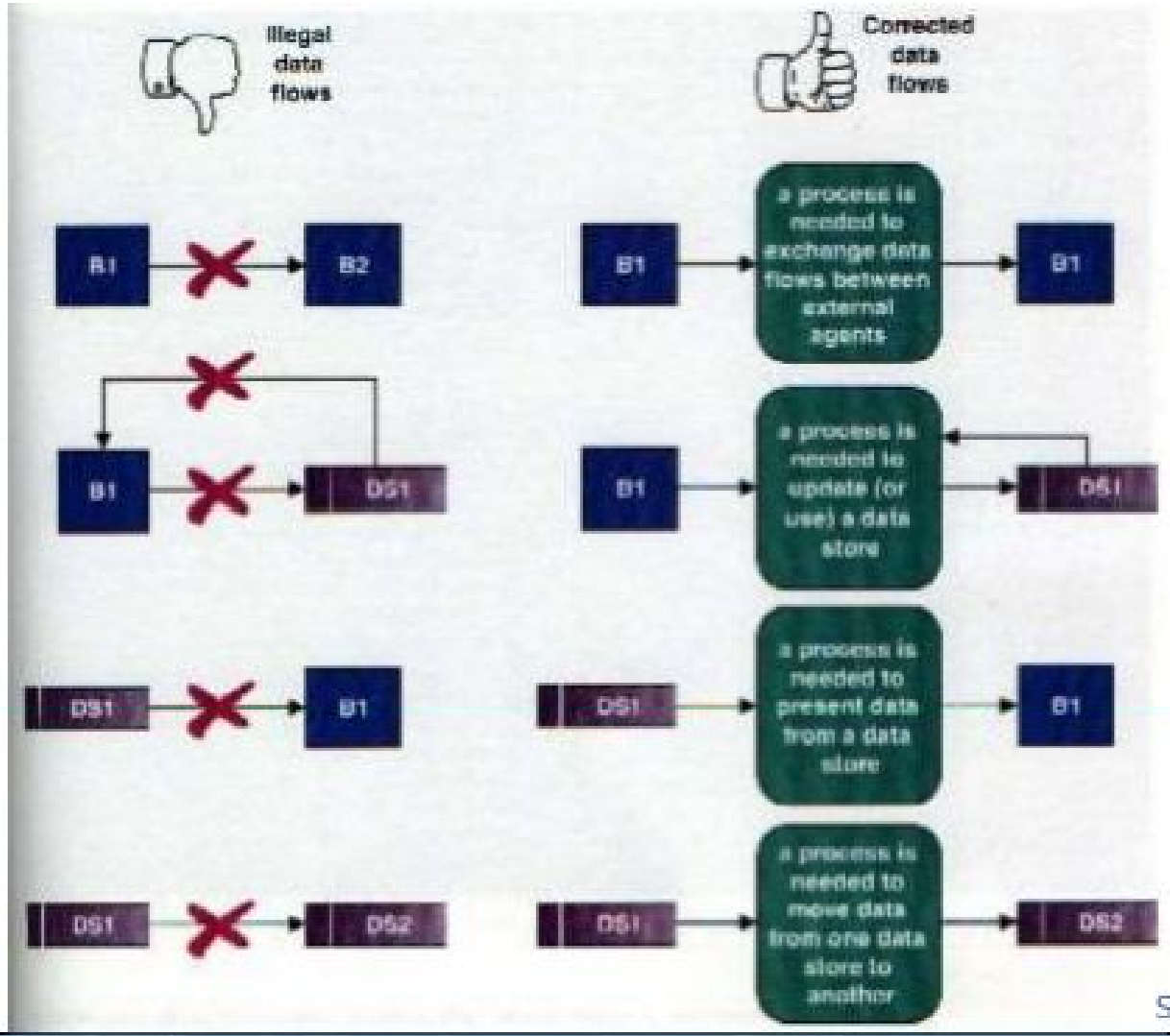
DFD Example


CUSTOMER
1.0
Record Payment
Payment
Receipt
2.0
Make Bank Deposit
Payment Data
3.0
Update Customer Master
Payment Data
Deposit Data
BANK
Credit Data
D1
CUSTOMER MASTER

Common Rules :

1. All processes must have at least one data flow in and one data flow out.
2. All processes should modify the incoming data, producing new forms of outgoing data.
3. Each data store must be involved with at least one data flow.
4. Each external entity must be involved with at least one data flow.
5. A data flow must be attached to at least one process.
6. In DFDs, all arrows must be labeled.

Common errors in DFD



Rules of Data Flow

- Data can flow from
  - ✓ External entity to process
  - ✓ Process to external entity
  - ✓ Process to store and back
  - ✓ Process to process
- Data cannot flow from
  - External entity to external entity
  - External entity to store
  - Store to external entity
  - Store to store

Bus Garage Repairs (cont'd)

- ***External Entities:***

  Bus, Mechanic, Helper, Supervisor, Inventory Management System, Accounting Department

- ***Key process*** ("the system"): performing repairs and storing information related to repairs.
- ***Processes:***

  Record Bus ID and reason for repair, Determine parts needed, Perform repair, Calculate parts extended and total cost, Record labor hours, cost

- ***Data stores:***

  Personnel file, Repairs file, Bus master list, Parts list

- ***Data flows:***

  Repair order, Bus record, Parts record, Employee timecard, Invoices

External Entities

- An External Entity is a provider (source) or receiver (sink) of **data and information** of the system

External Entities …

- 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 system.
  - One of system's end-users or managers

Data Stores

- A data store is an inventory of data.
  - A data store means “data at rest.”
  - A data flow means “data in motion.”

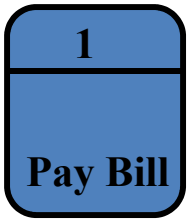
Data Stores …

- **Almost always one of the following:**
  - Persons (or groups of persons): e.g., customer
  - Places: e.g, cash register
  - Objects: e.g., product
  - Events (about which data is captured): e.g., sales

Data Flows

- A Data Flow represents an input of data **to a process**, or the output of data **from a process**.
- 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

Processes



- A Process is a **work** or **action** performed on input data flow to produce an output data flow
- Use a verb/verb phrase to label the action performed by the process
- A Process must have **at least one input** data flow and **at least one output** data flow.

How to find each one…

- External Entities (EE): noun
  - people/organizations/things that supply information to or use information from the system
- Processes (P): verb/verb phrase
  - actions/doing words
- Data Flows (DF): name of data
  - movement/exchange of information/data between external entities to processes, and processes to processes
- Data Stores (DS): noun
  - store/record information/data

Example for finding each one…

- A student (EE/DS) sends in an application form (DF) containing their personal details, and their desired course
- The university (EE) checks (P) that the course (EE/DS) is available.
- If the course is available and the student is enrolled (P) in the course, the university confirms (P) the enrolment by sending a confirmation letter (DF) that they are registered (P) for the course to the student.
- Or if the course is unavailable the student is sent a rejection letter (DF).
- What system is being modeled?

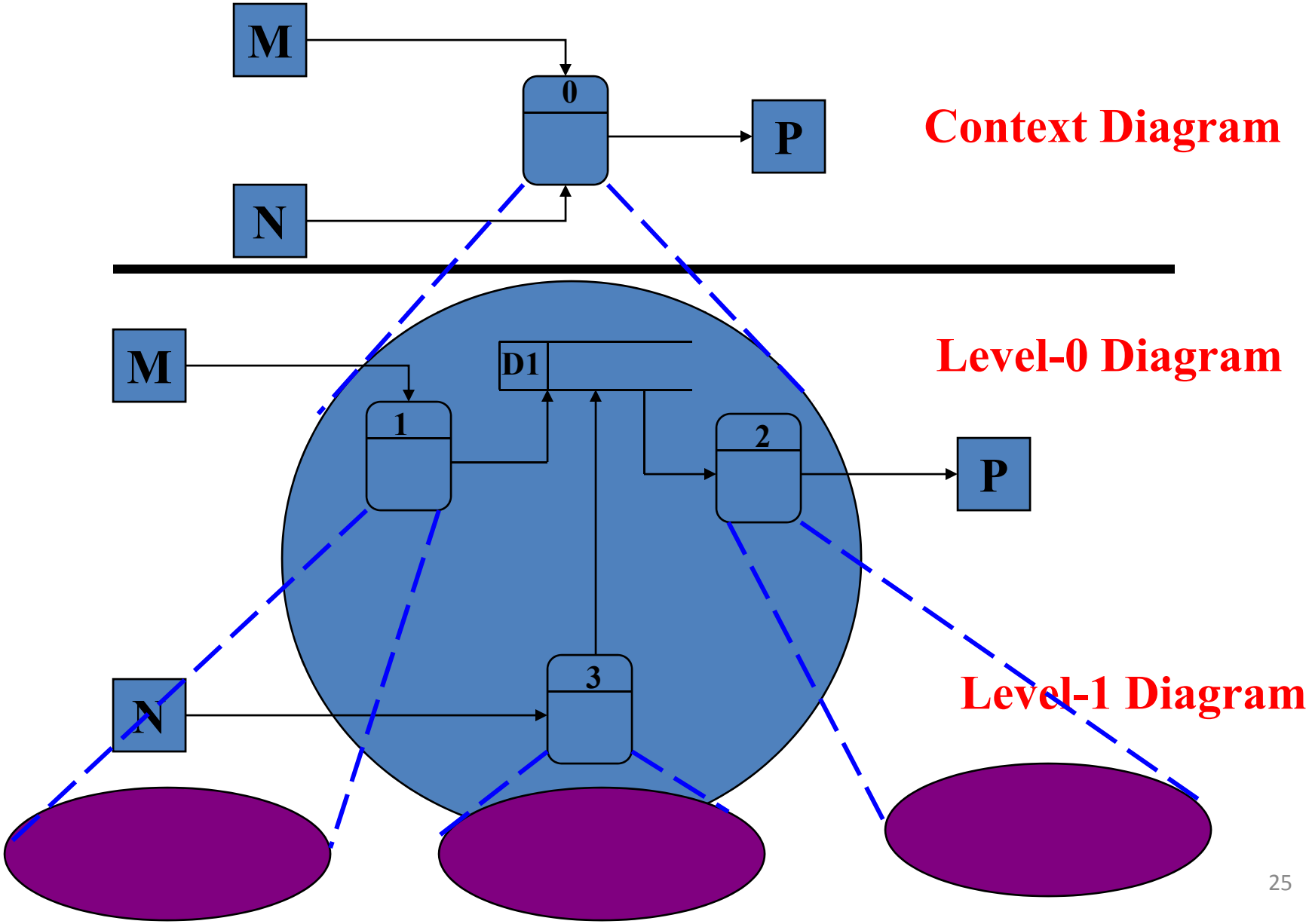
Guidelines for Developing DFD

1. Develop a list of business activities
   - Similar to business rules of ERD
2. Create the Context Diagram **(first task)**
   - shows **external entities** and **data flows** to and from the system: conceptual DFD
3. Create level 0 DFD (or Diagram 0)
   - shows general processes at the highest level
4. Create level 1 DFD (or Child Diagram)
   - shows more detailed processes
5. Create level n-1 DFD…..

Context Diagram Building Procedure

- Identify the system and its boundaries (the context)
- Identify external entities (providers, receivers of system info)
- Identify external data flows (input, output)
  - **However, NO DATA STORE !!!**
- Note: the whole system itself is a process (it receives input and transforms into output)
- **Go back to the class website for the Context Diagram!**

Decomposition of Context Diagram



Description of Each Level

- Context Diagram: This is the highest level and represents the overall system and its interaction with its environment
- Level 0 Diagram: This shows the major subsystems and their interactions
- Level x Diagram: Shows the processes that make up each of the major subsystems

Level 0 DFD Building Procedure

- Level-0 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**

Level 0 Tips

- Generally move from top to bottom, left to right
- Minimize crossed lines
- Iterate as needed
  - *The DFD is often drawn many times before it is finished,* ***even with very experienced systems analysts***

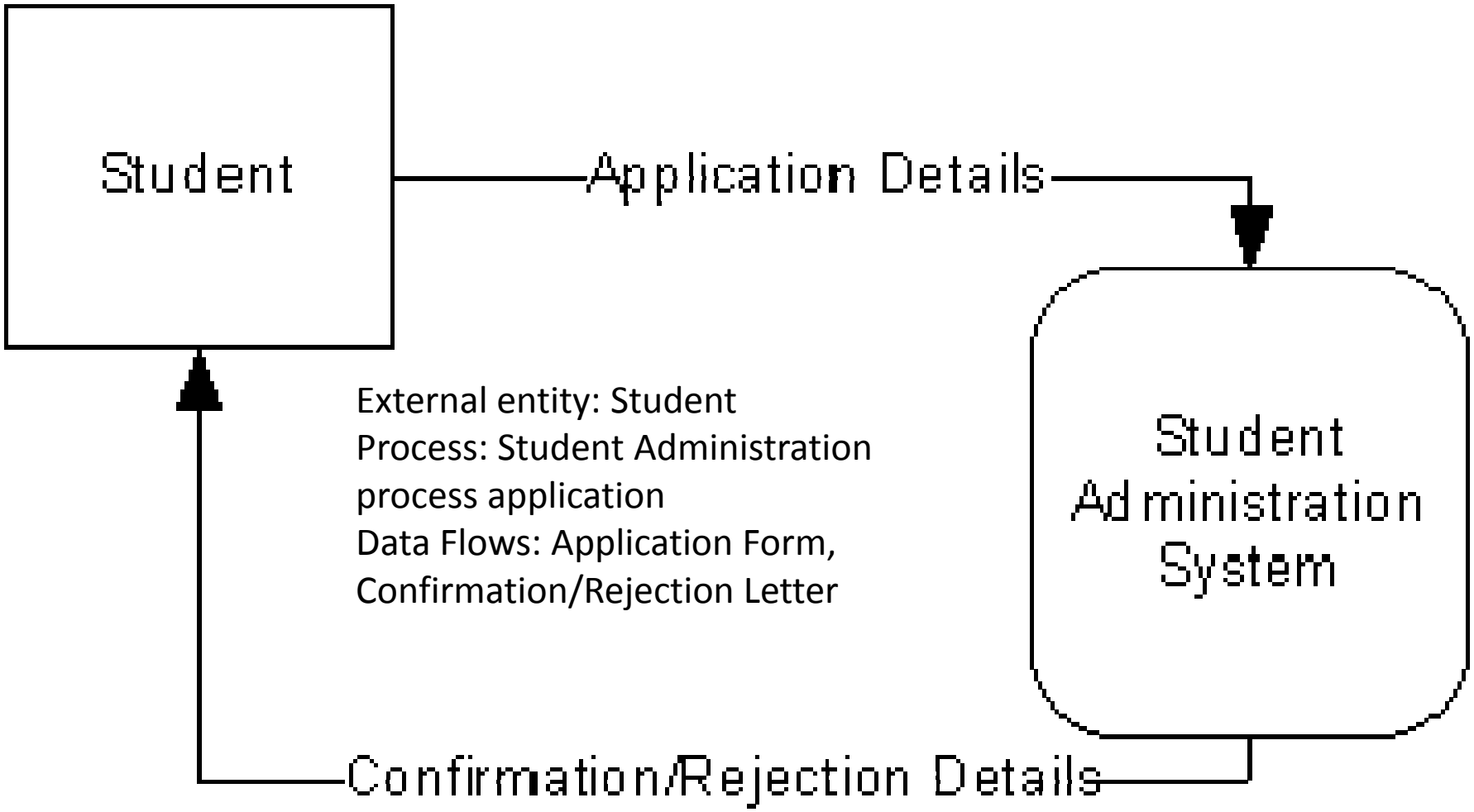
Tips for Level 1 and Below

- Sources for inputs and outputs listed at higher level
- List source and destination of data flows to processes and stores within each DFD
- Depth of DFD depends on overall system complexity
  - Two processes generally don't need lower level
  - More than seven processes become overly complex and difficult to read

Tips for Level 1 and Below

- Level-1 DFD's
  - Sub-processes (primitive processes) of Level-0 processes
- Sources for inputs and outputs listed at higher level
- List source and destination of data flows to processes and stores within each DFD
- Depth of DFD depends on overall system complexity
  - Two processes generally don't need lower level
  - More than seven processes become overly complex and difficult to read

Context diagram of Student Administration System



External entity: Student
Process: Student Administration process application
Data Flows: Application Form, Confirmation/Rejection Letter

Apply logical sequence of the system..

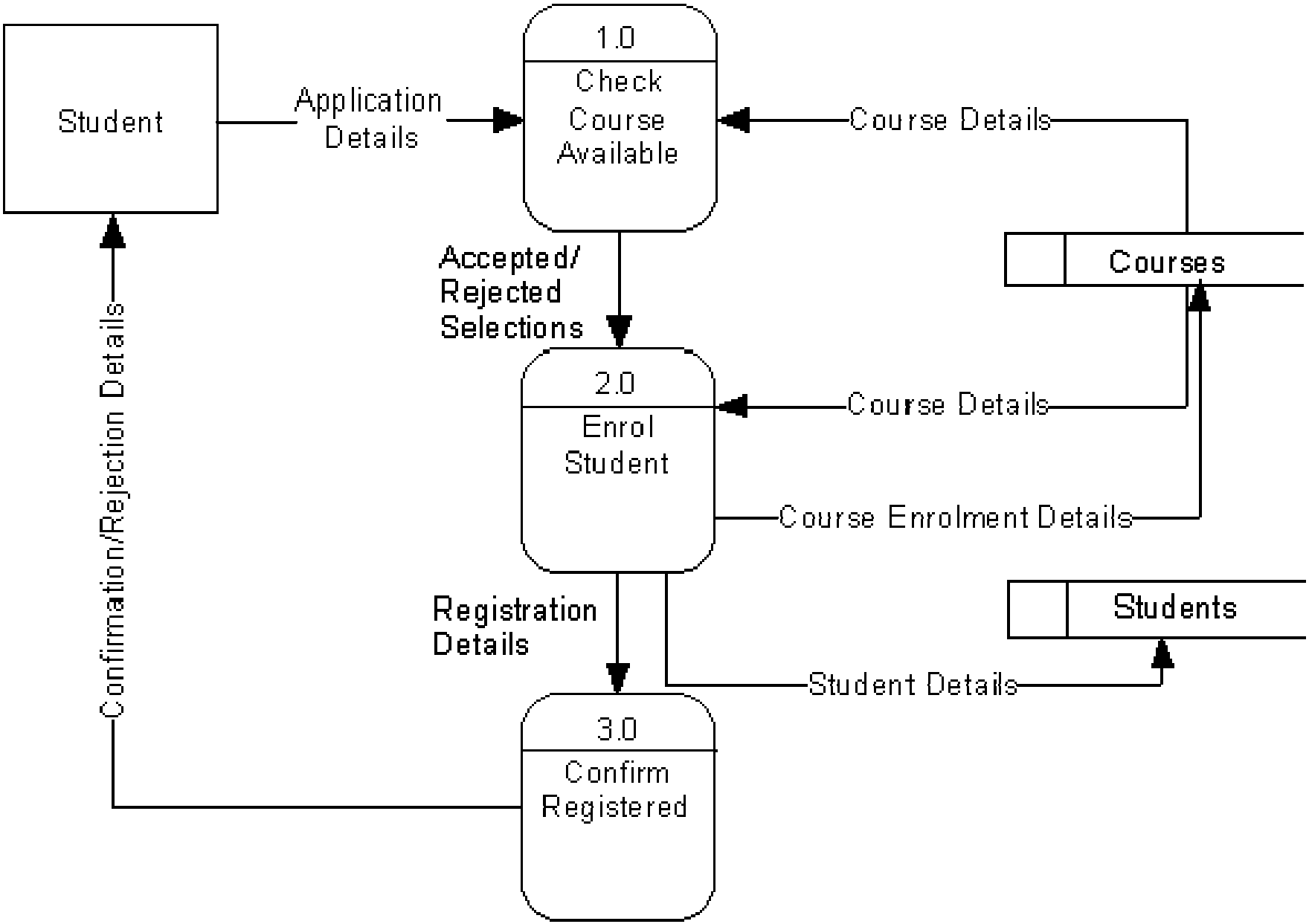
- A student (EE / DS) sends in an application form (DF) containing their personal details, and their desired course
- The university checks (P) that the course is available.
- If the course (DS) is available the student is enrolled (P) in the course, and the university confirms (P) the enrolment by sending a confirmation letter (DF) that they are registered (P) for the course to the student.
- Or if the course is unavailable the student is sent a rejection letter (DF).

How to find each one…

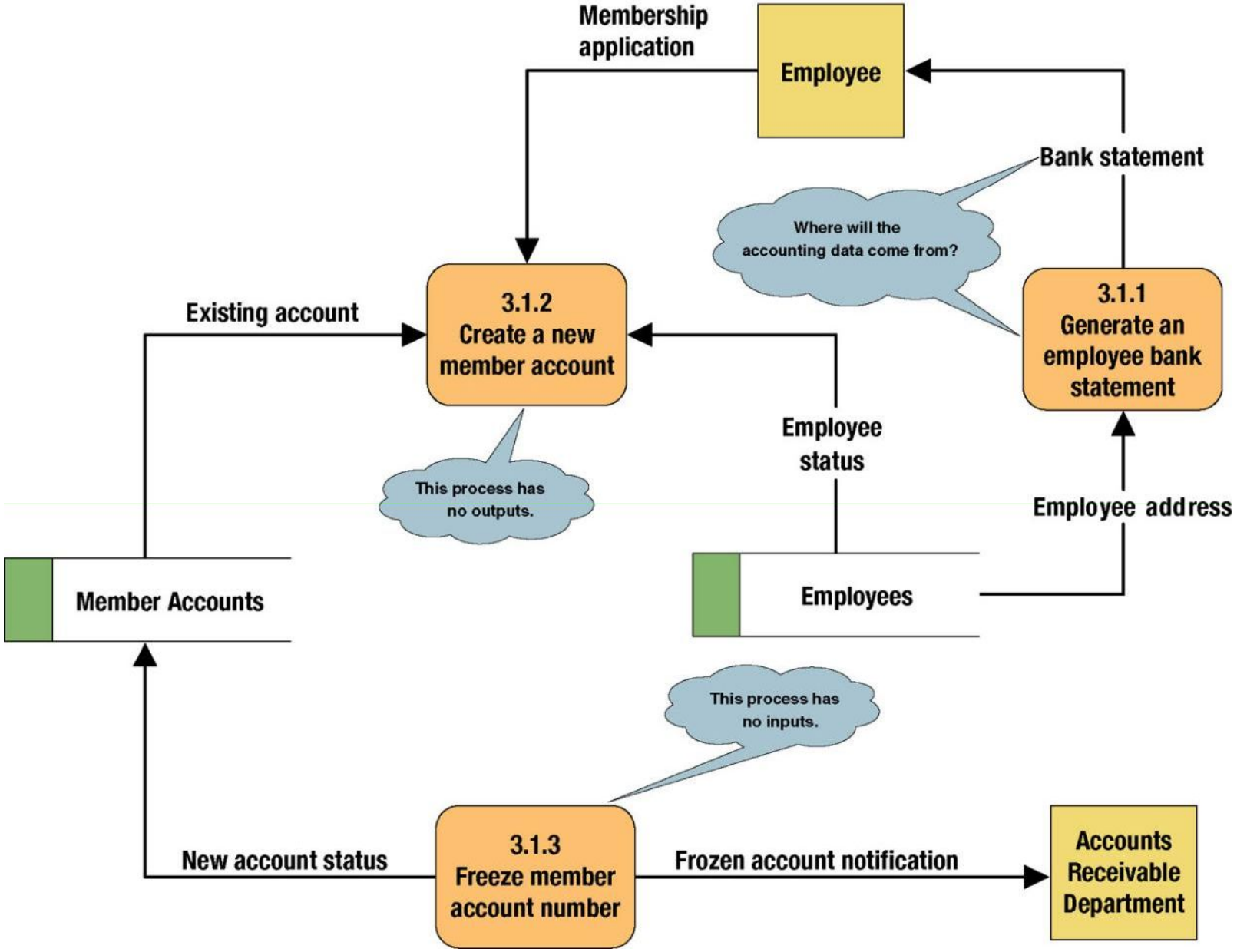
- External Entities (EE): noun
  - people/organizations/things that supply information to or use information from the system
- Processes (P): verb/verb phrase
  - actions/doing words
- Data Flows (DF): name of data
  - movement/exchange of information/data between external entities to processes, and processes to processes
- Data Stores (DS): noun
  - store/record information/data

Level-0 DFD

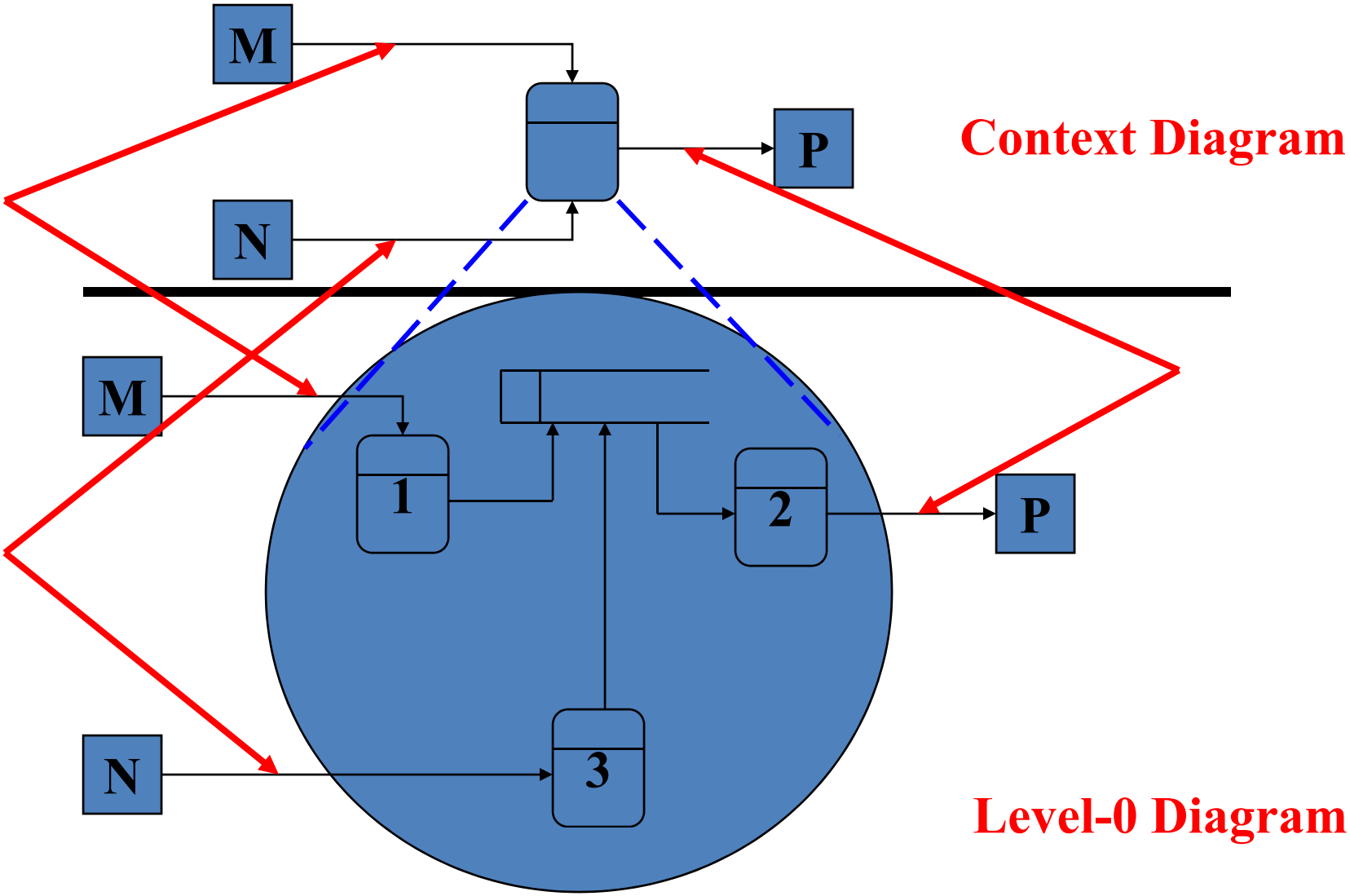
- External entity: Student
- Processes: Check course available, Enroll student, Confirm Registration
- Data Flows: Application Form, Course Details, Course Enrolment Details, Student Details, Confirmation/Rejection Letter
- Data Stores: Courses, Students.
- Also see the “Lemonade Stand” example from the class website


Student
Application Details
1.0
Check Course Available
Course Details
Courses
Accepted/ Rejected Selections
2.0
Enrol Student
Course Details
Course Enrolment Details
Confirmation/Rejection Details
Registration Details
Students
Student Details
3.0
Confirm Registered

Common Process Errors on DFDs



Rules in DFD Building ...



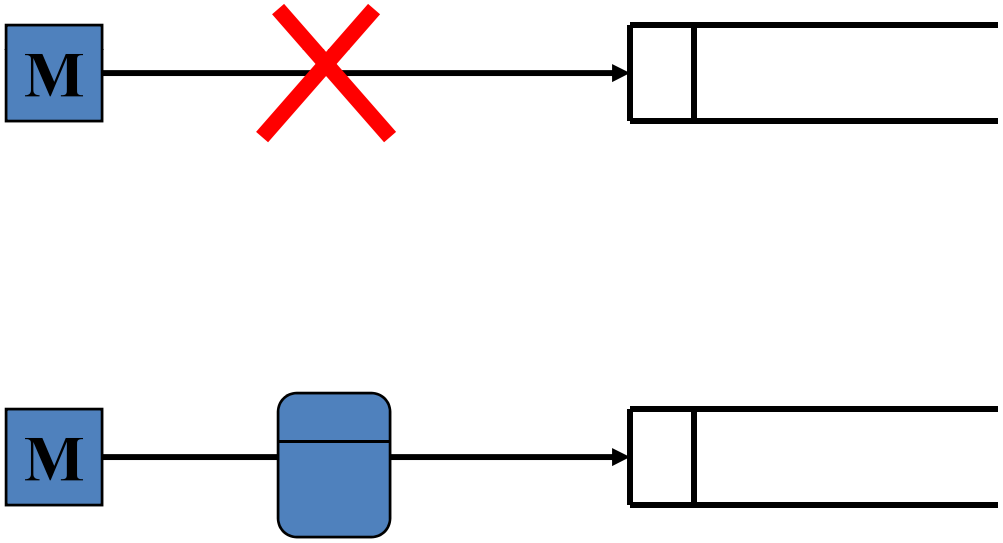
Rules in DFD Building

- Rule 1 : Use only DFD notations to avoid confusion

- Rule 2 : Use an action VERB to label a process (because a process is an action !!!)

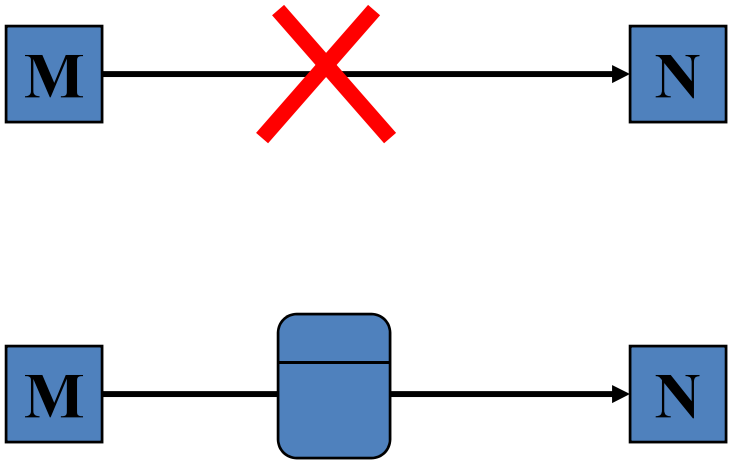
Rules in DFD Building ...

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



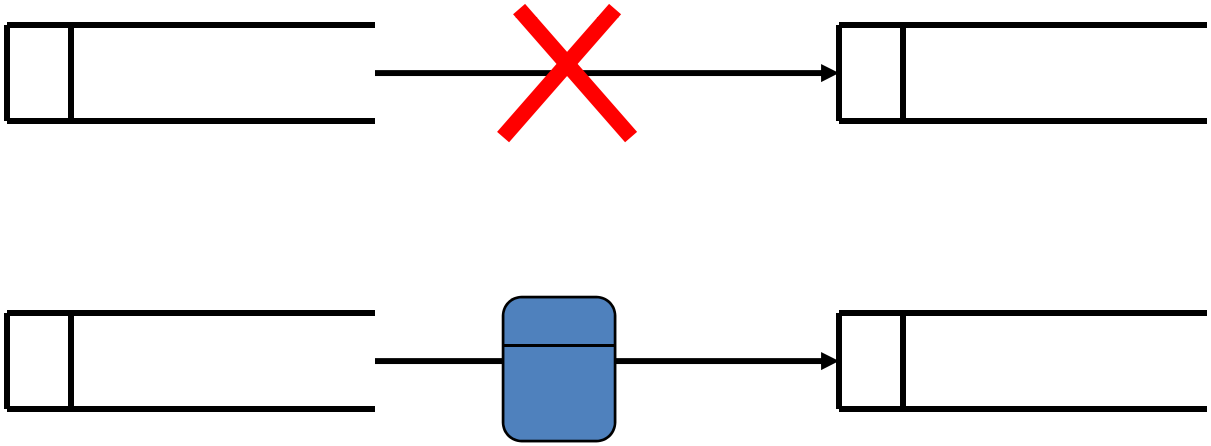
Rules in DFD Building ...

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



Rules in DFD Building ...

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

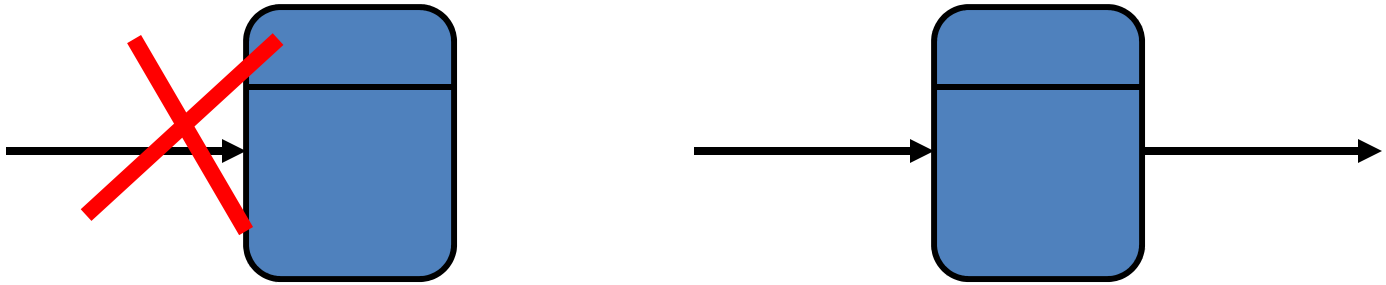


Rules in DFD Building ...

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

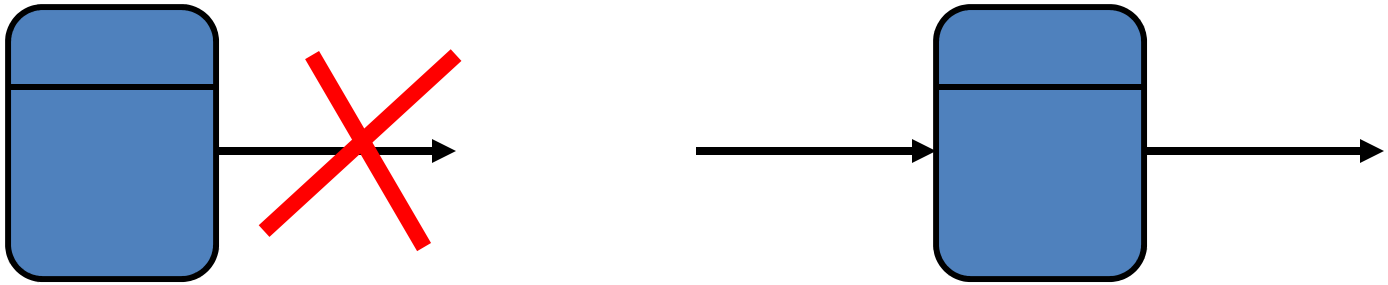
Rules in DFD Building ...

- Rule 5 : No process without output data flow (black hole !!!)



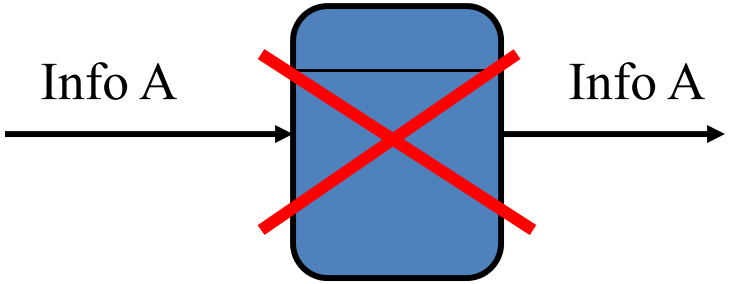
Rules in DFD Building ...

- Rule 6 : No process without input data flow (miracle !!!)



Rules in DFD Building ...

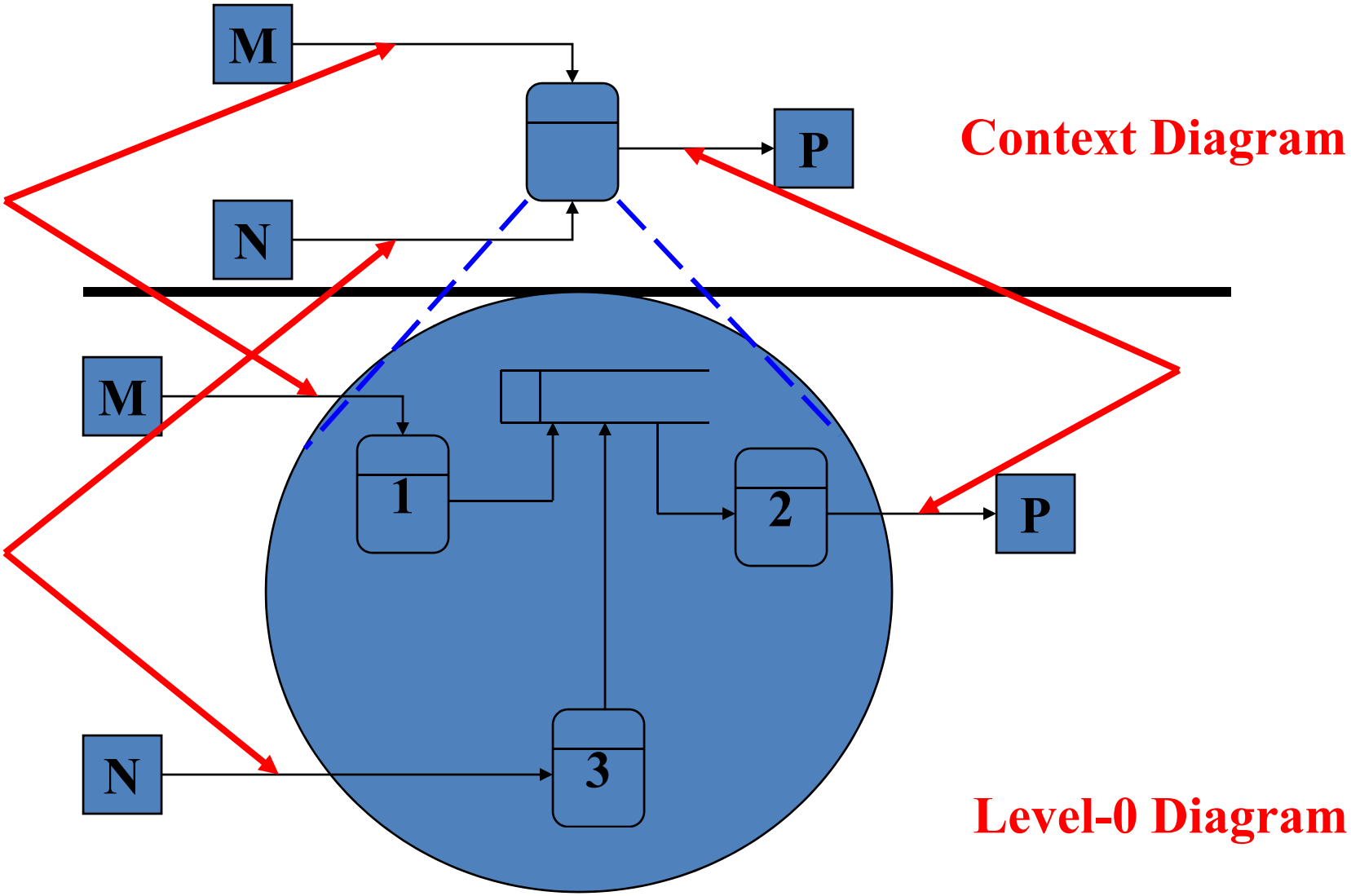
- Rule 7 : No need for routing (without transforming) a data flow with a process (non value-added activities !!!)



Rules in DFD Building ...

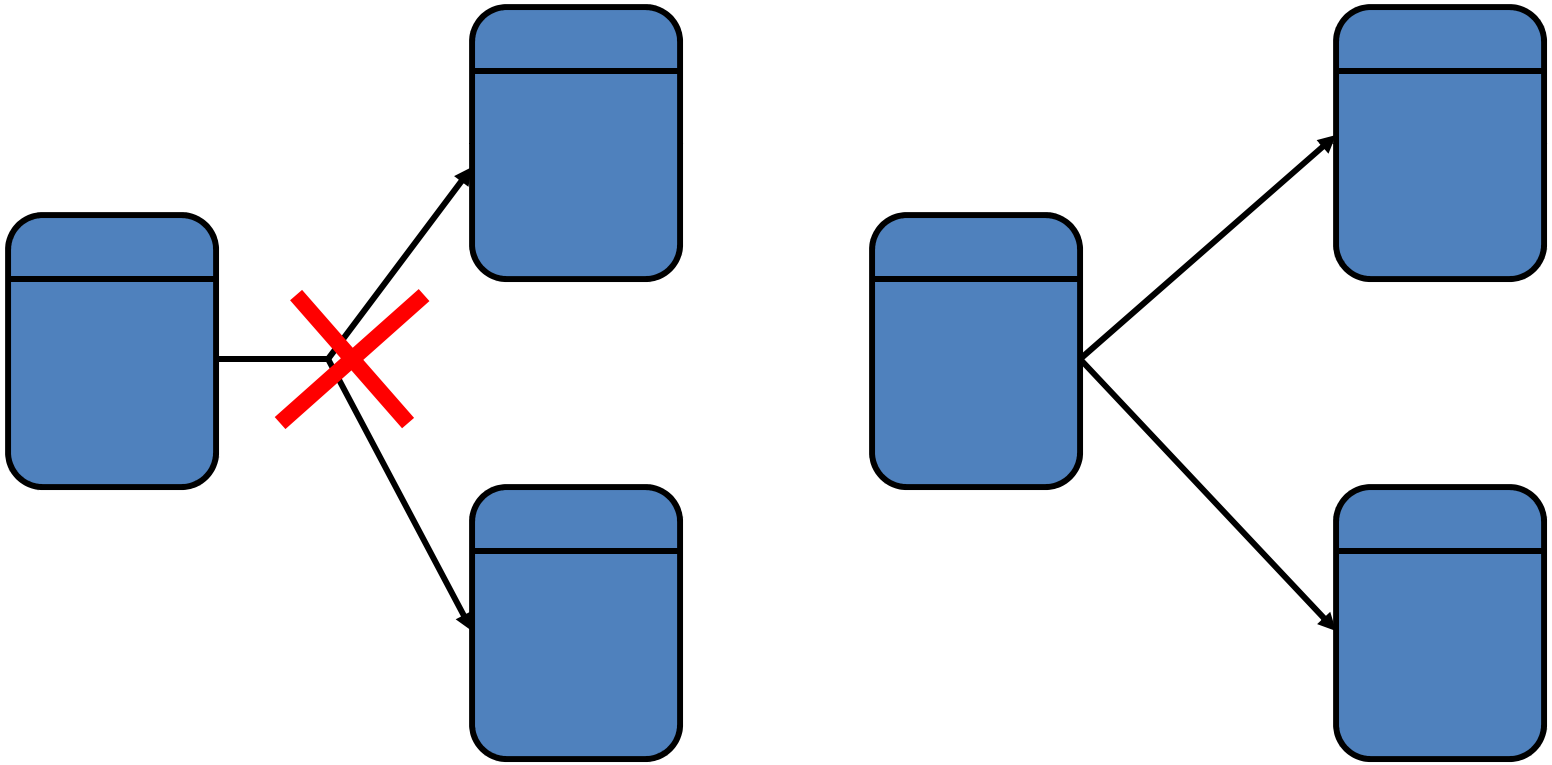
- Rule 8 : Identical input, output data flows for parent and child processes (but the child processes can have their own throughputs)
  - See the picture in next slide

Detail Rules in DFD Building ...



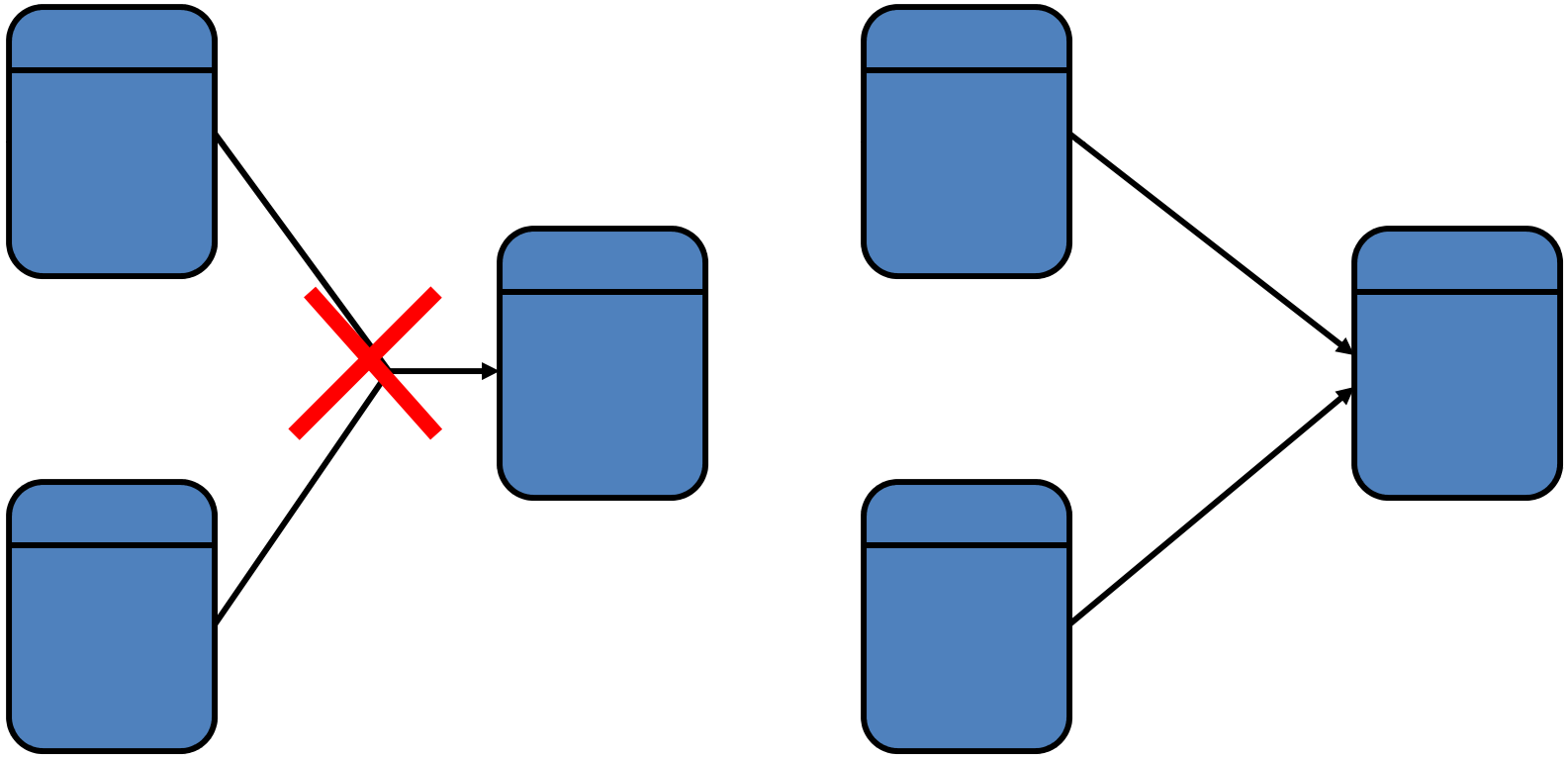
Rules in DFD Building ...

- Rule 9 : Data flows cannot split by themselves



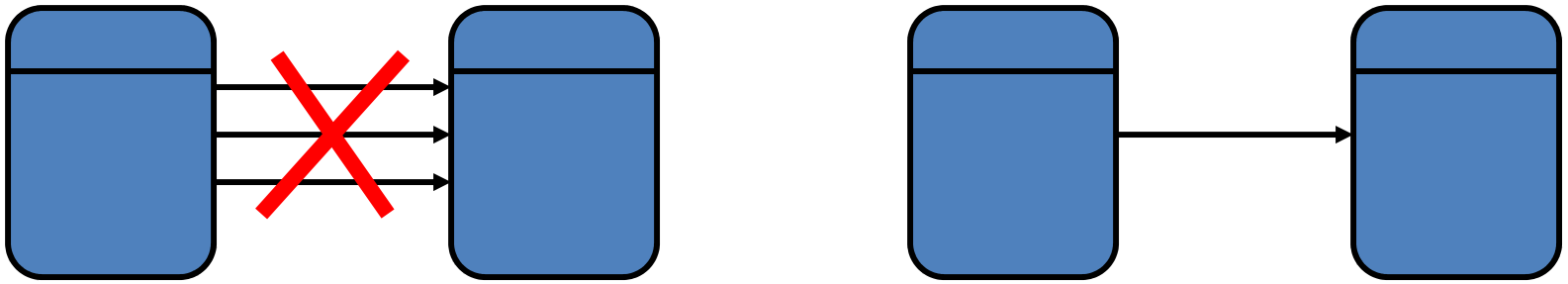
Rules in DFD Building ...

- Rule 9 : Data flows cannot split …



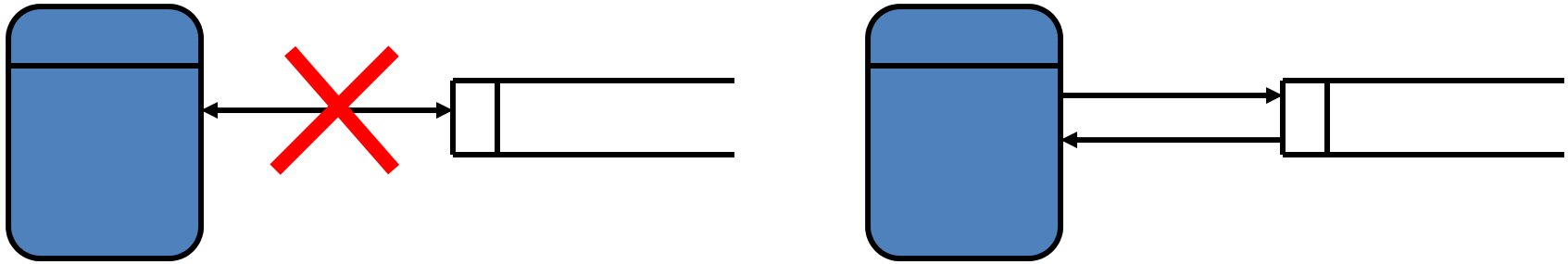
Rules in DFD Building ...

- Rule 10 : A data packet can combine many data elements being transmitted *at the same time* to the same destination

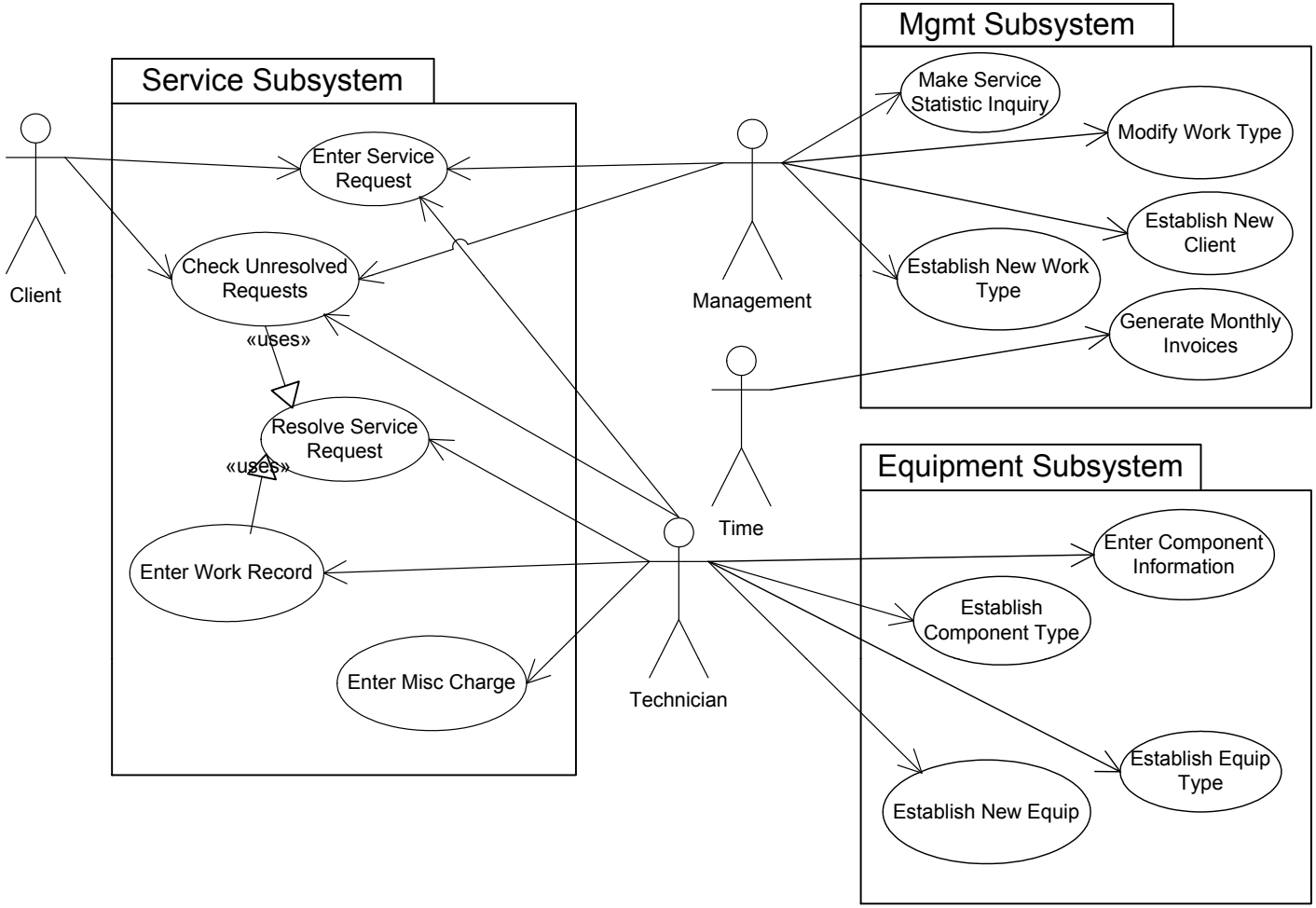


Rules in DFD Building ...

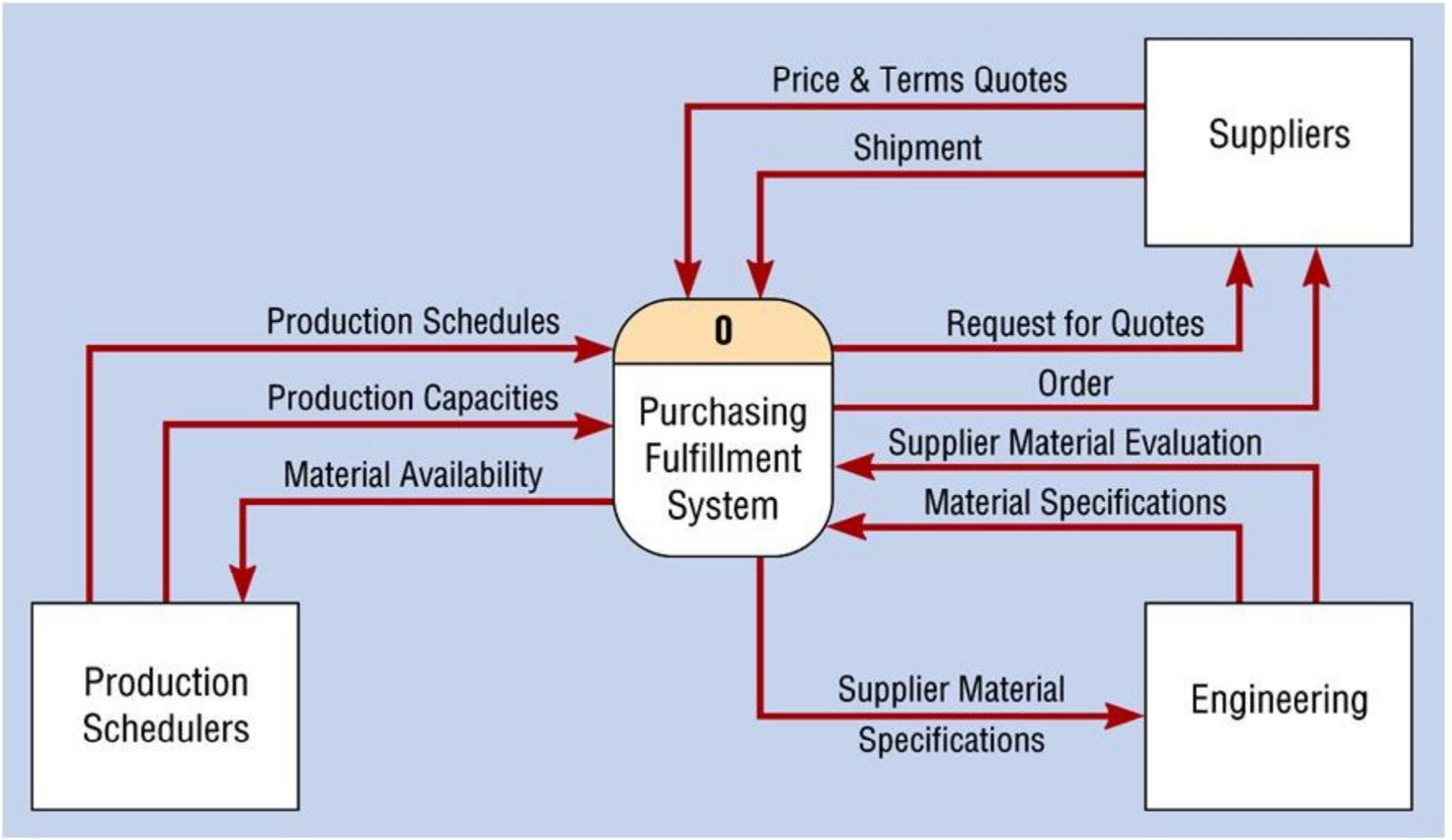
- Rule 11 : Double-headed arrows are forbidden [in- flow (update) and out-flow (extract info) of a data store are different]



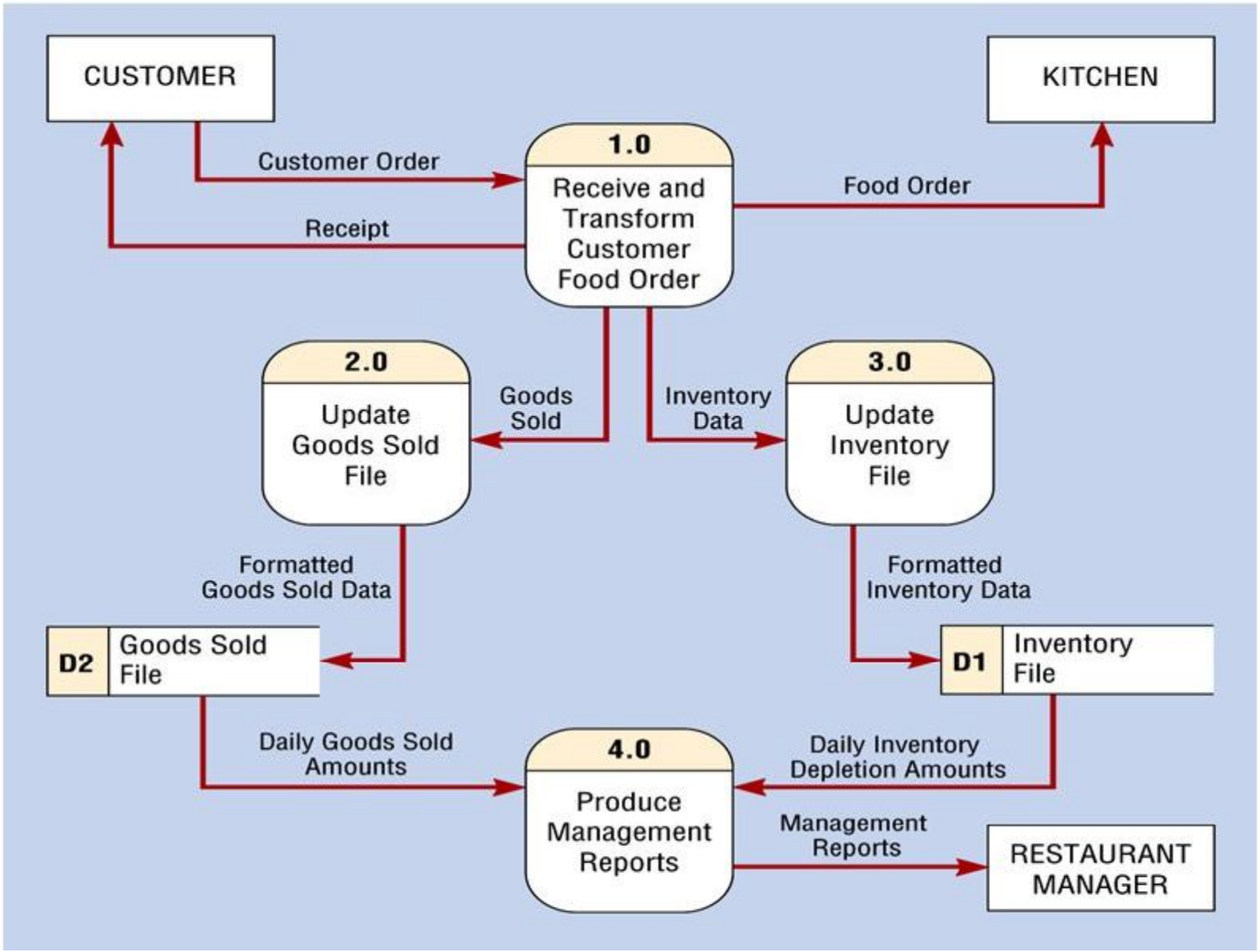
Decomposition: Use-Case Diagram

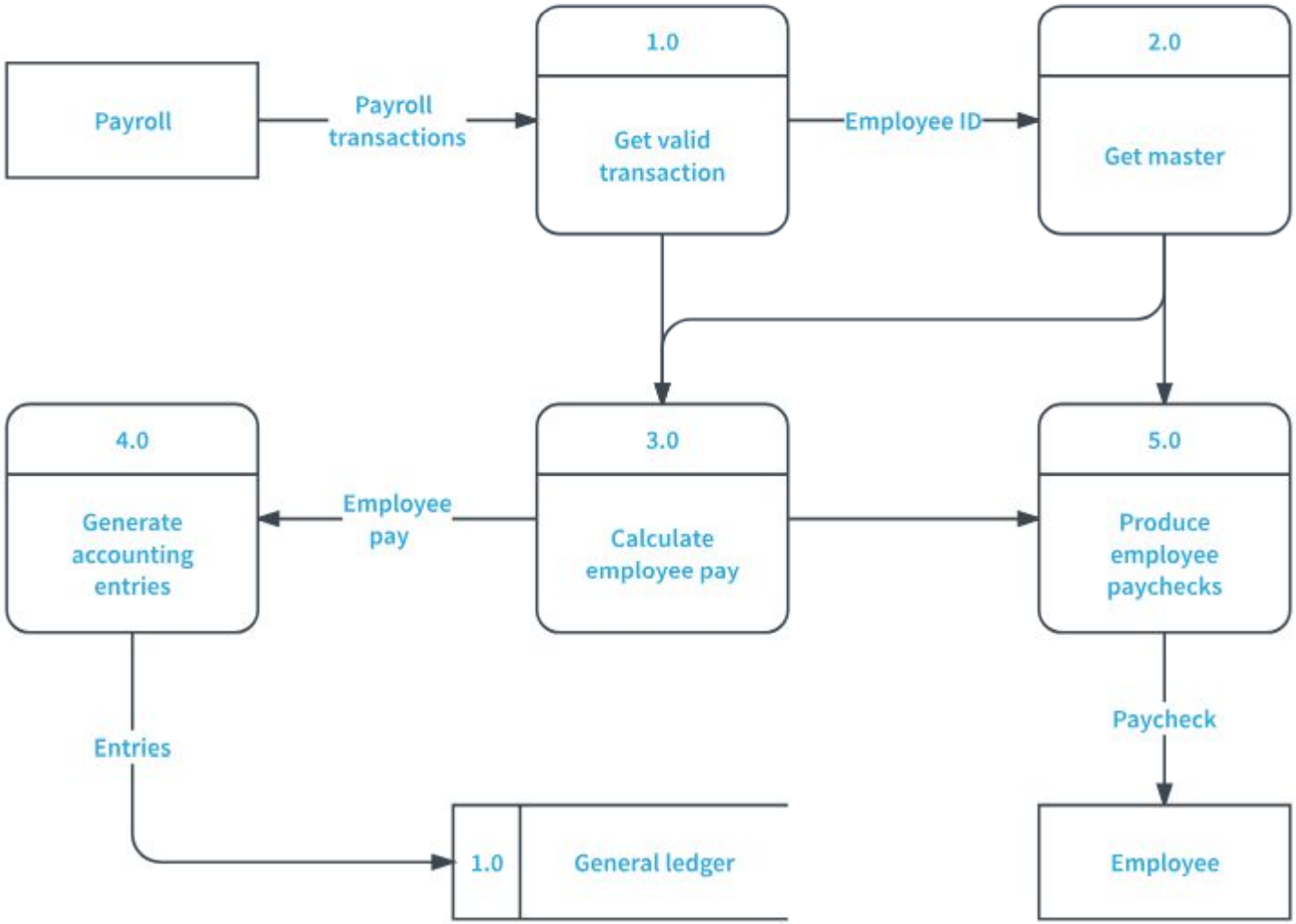


Context-Level Data Flow Diagram Showing Project Scope for Purchasing Fulfillment System



Level-0 DFD of Hoosier Burger's Food Ordering System




Payroll
Payroll transactions
1.0
Get valid transaction
Employee ID
2.0
Get master
4.0
Generate accounting entries
Employee pay
3.0
Calculate employee pay
5.0
Produce employee paychecks
Entries
1.0
General ledger
Paycheck
Employee