SOFTWARE ENGINEERING

Week 6

Requirements Analysis Model

Assoc. Prof. Dr. Ayşe Tosun Tantuğ

Assoc. Prof. Dr. Cüneyd

Agenda

1. Requirements Analysis
2. Structured Analysis
   1. Data Model: Database objects and relations
   2. Functional Model: Data flow
   3. Behavioural Model: Control flow, Events and states

1. Requirements Analysis
2. Structured Analysis
    1. Data Model
    2. Functional Model
    3. Behavioural Model

Requirements Analysis

6.1

Analysis and Design Approaches

Structured Analysis and Design

Data / Control Flow Diagrams, (DFD / CFD)

Object-Oriented Analysis and Design

Unified Modeling Language Diagrams (UML)

Elements of Analysis Model

- The Statement of Software Scope provides the basis for analysis modelling.
- The following models are built during analysis:

1. Data model: Database objects and relations
2. Functional model: Data flow
3. Behavioural model: Control flow, Events and states

Modeling the Data Domain

- Define data objects
- Establish data relationships
- Specify data content

Modelling the Functions

- Basic Idea:
  - Software transforms data
  - To achieve this, it must perform at least three generic functions: input, processing, output
  - Identify functions that transform data objects
- Begin with a context level diagram (level 0)
- Continue with more functional details in refined levels until all system functionality is represented

Modeling the Behaviour

- Basic Idea:
  - Most software responds to events from the outside world
  - This characteristic forms the basis of the behavioral model
  - A computer program always exists in some state: an externally observable mode of behaviour (e.g. waiting, computing, printing, polling) that is changed only when some event occurs
- Indicate different states of the system
- Specify events that cause the system to change state

1. Requirements Analysis
2. Structured Analysis
    1. Data Model
    2. Functional Model
    3. Behavioural Model

Structural Analysis

6.2

1. Requirements Analysis
2. Structured Analysis
   1. Data Model
   2. Functional Model
   3. Behavioural Model

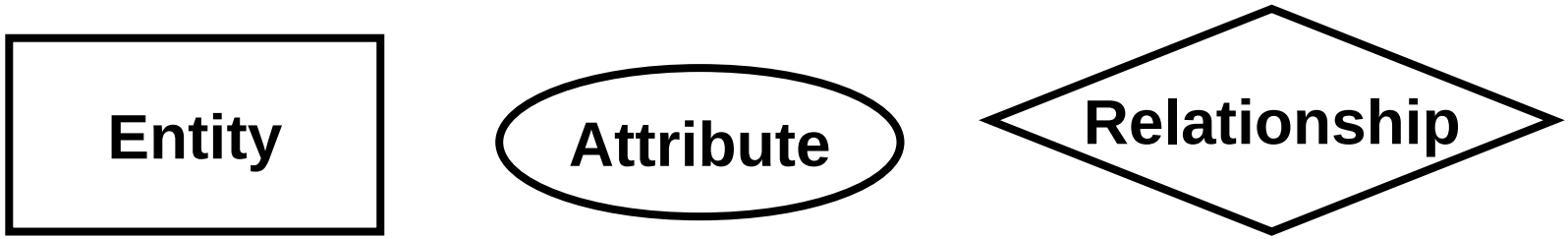
The Data Model

6.2.1

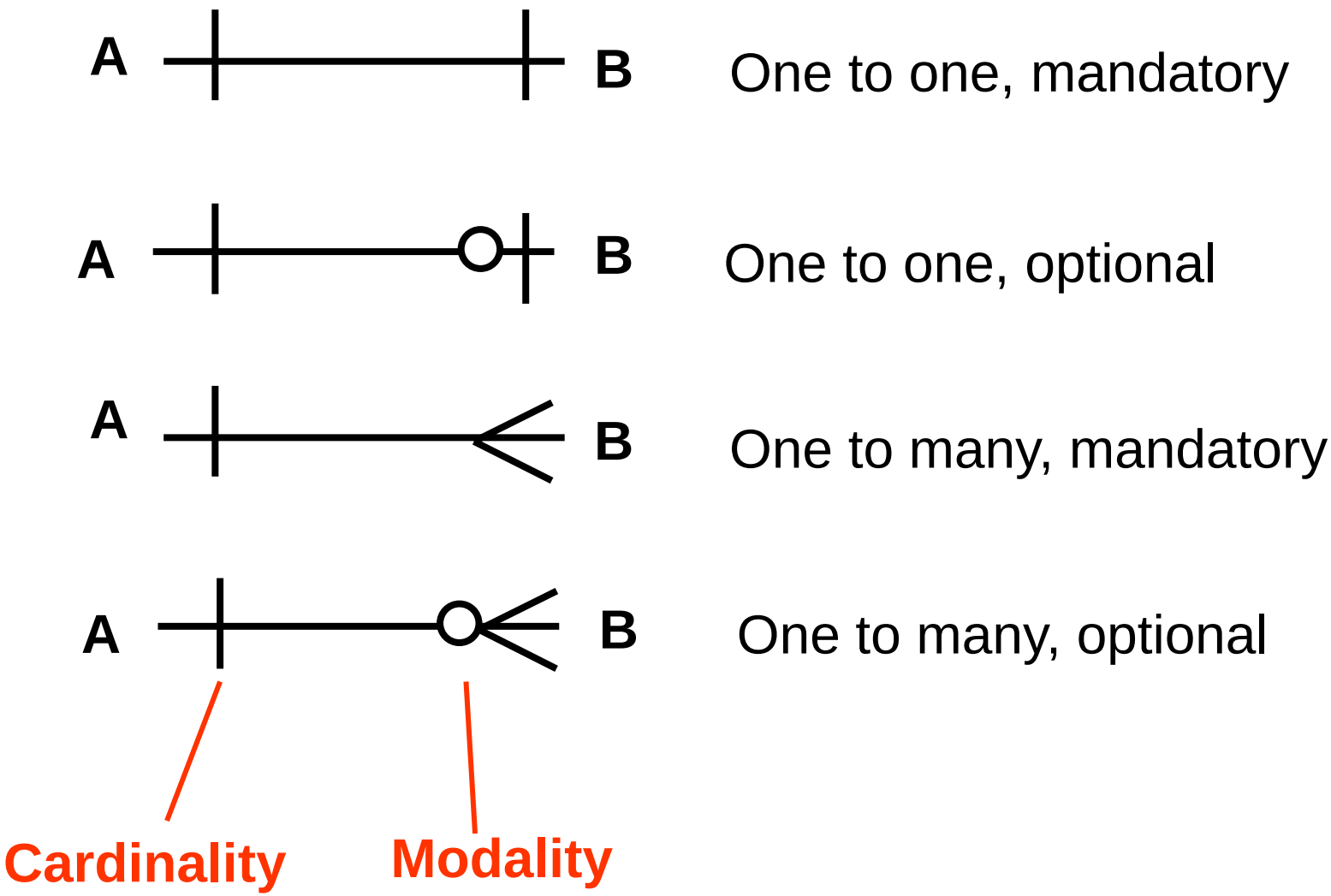
The Data Model

- Data modelling is also called Database Modelling.
- In data modelling, Entity-Relationship Diagrams are used.
- Also a data dictionary is defined for important data items.

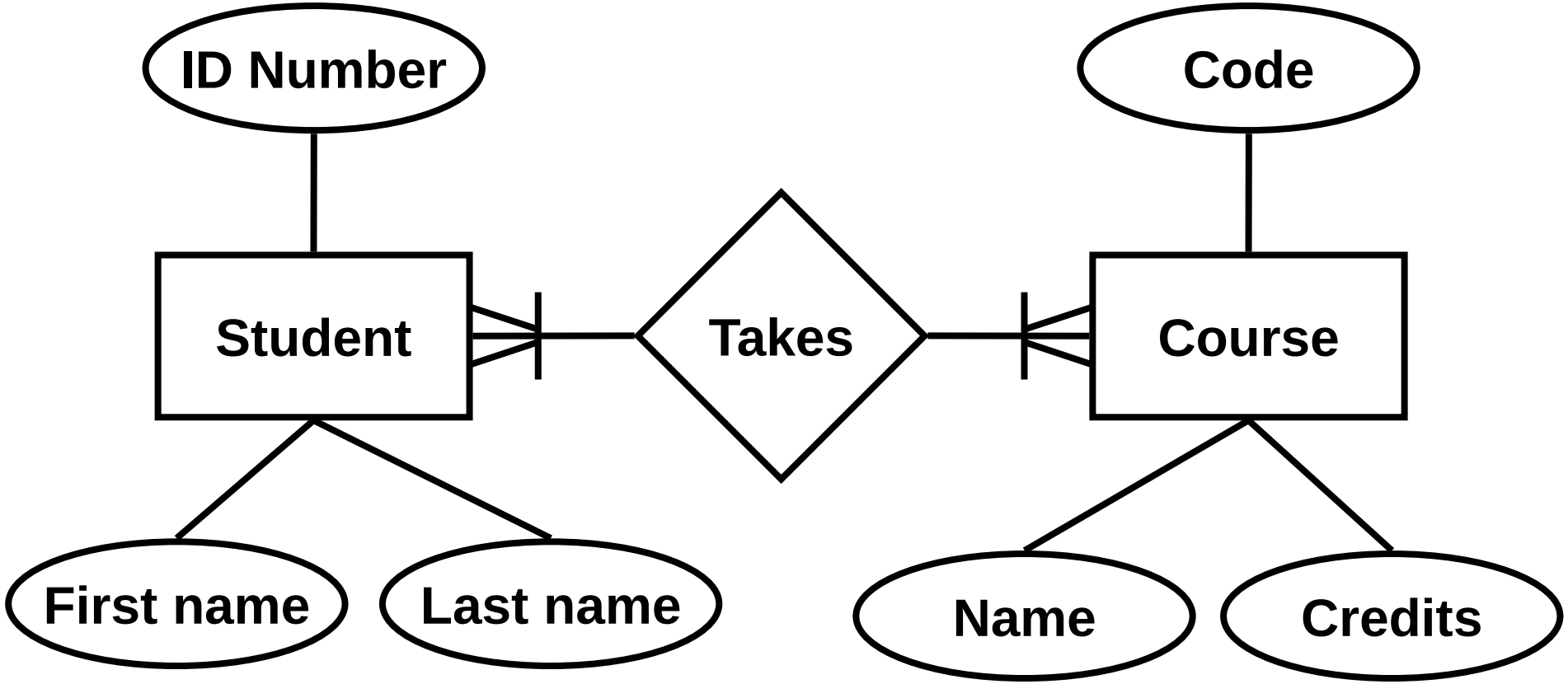
Entity Symbols (Bachman notation)



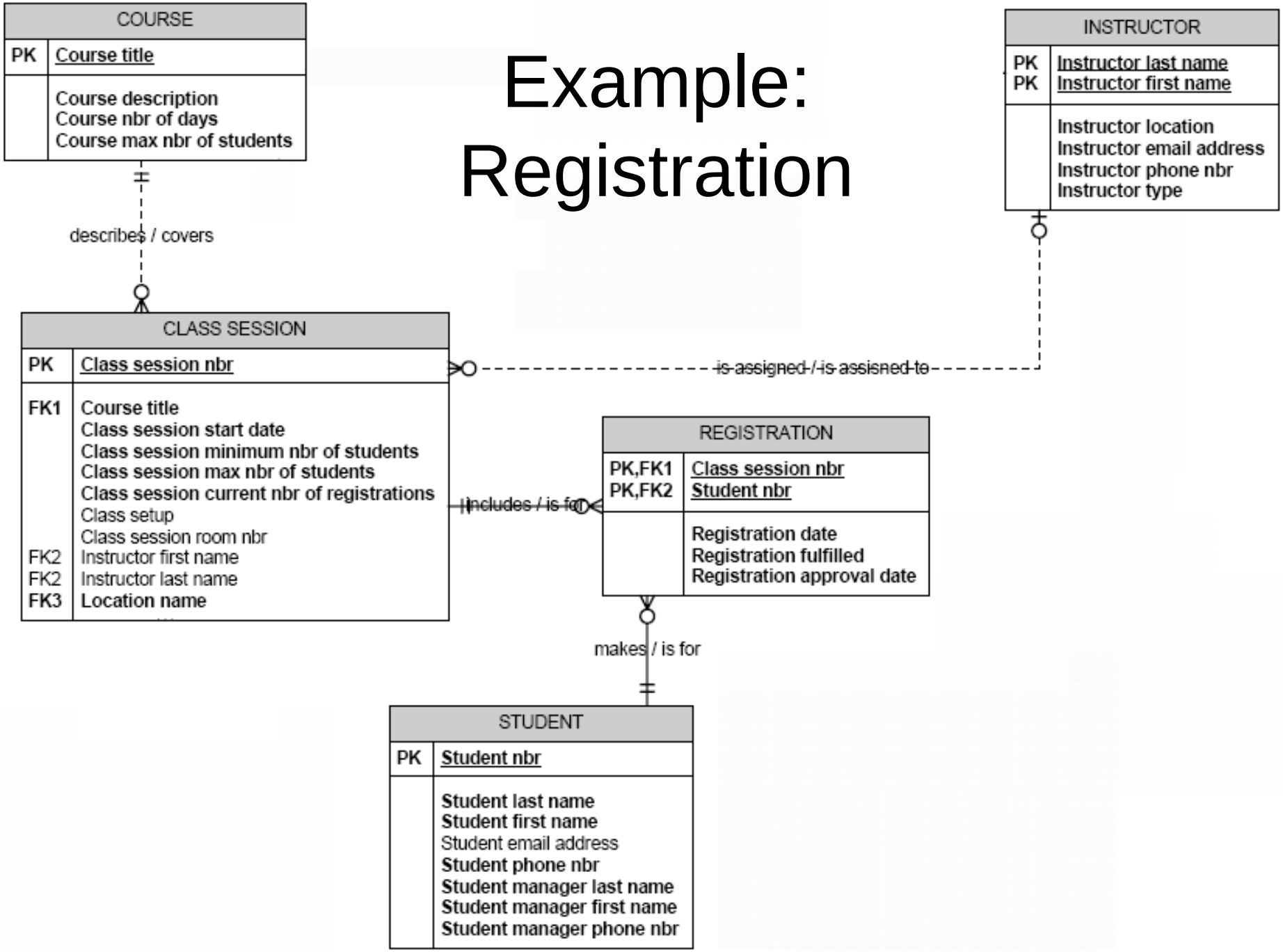
Relationship Symbols



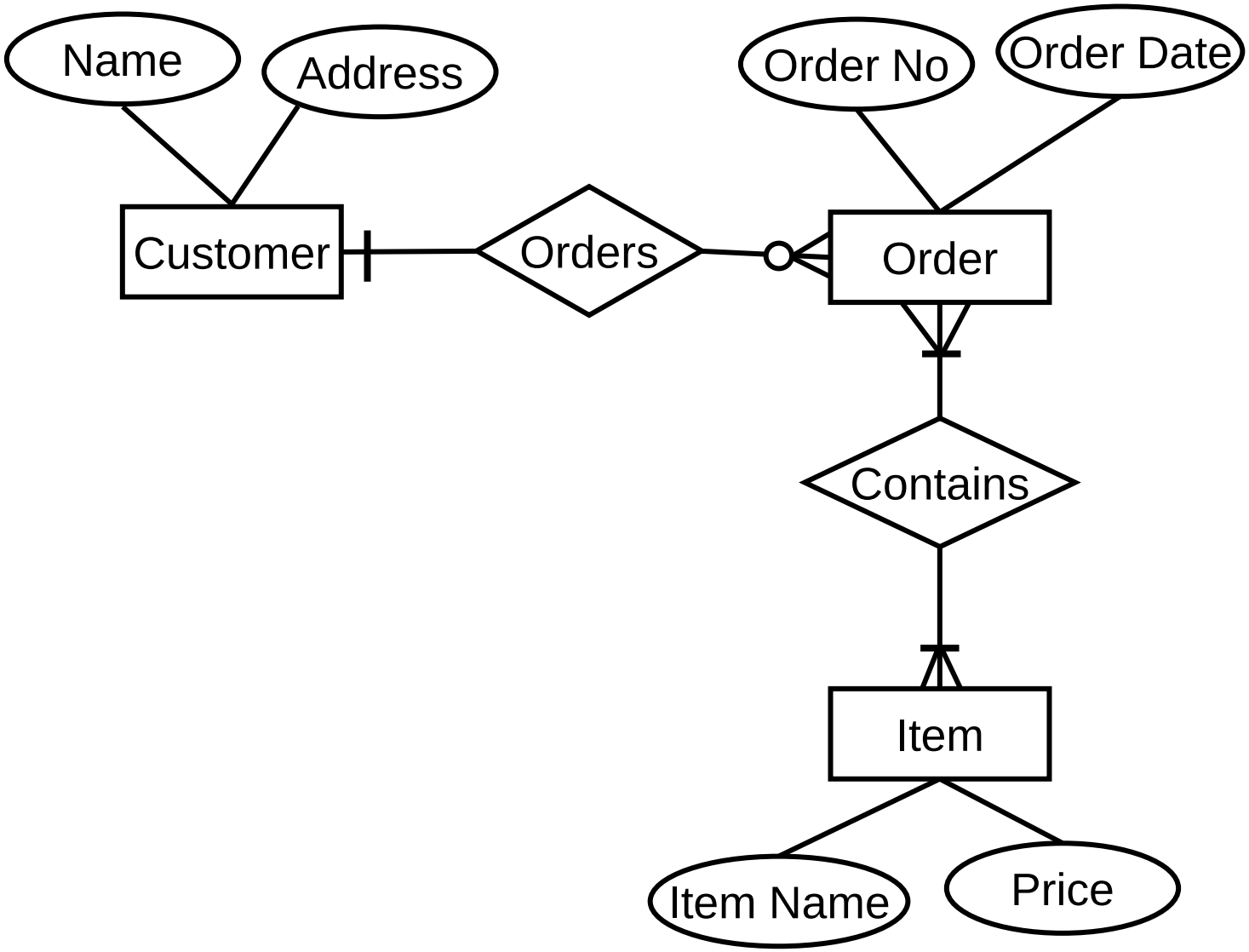
Example: Students and Courses (Bachman notation)



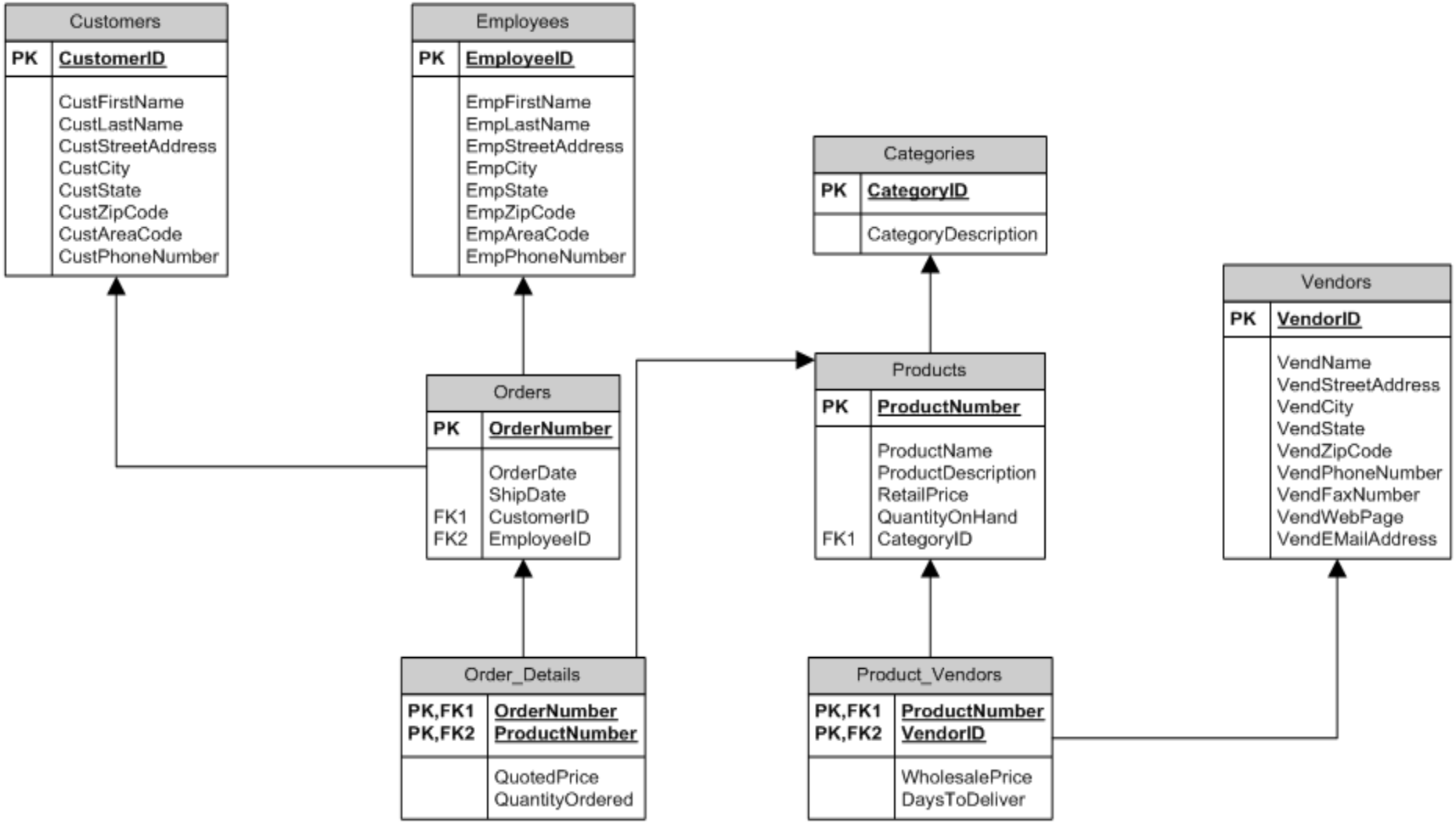
Example: Registration



Example : Orders



Example : Orders and Products



Data Dictionary

- Data dictionary is a collection of data item definitions.
- A data item is described with the followings:

Data Name	the primary name of the composite data item
Aliases	other names for the data item
Where used	data transforms (processes) that use the composite data item
How used	the role of the data item (input, output, temporary storage, etc.)
Description	a notation for representing content
Format	specific information about data types, default values (if known)

Data Dictionary Example

name:	telephone number
aliases:	none
where used/how used:	assess against set-up (output) dial phone (input)

description:

telephone number = [local number|long distance number]

local number = prefix + access number

long distance number = 1 + area code + local number

area code = [800 | 888 | 561]

prefix = *a three digit number that never starts with 0 or 1*

access number = * any four number string *

Data Dictionary Example

İTÜ Student ID = 04 – 009 - 0761

Electrical-Electronics faculty

Entrance year

Order number

1. Requirements Analysis
2. Structured Analysis
   1. Data Model
   2. Functional Model
   3. Behavioural Model

The Functional Model

6.2.2

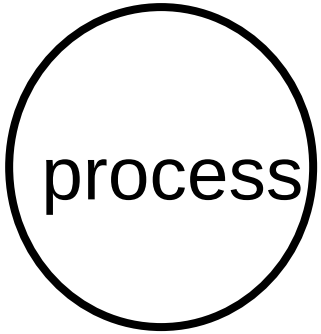
The Functional Model

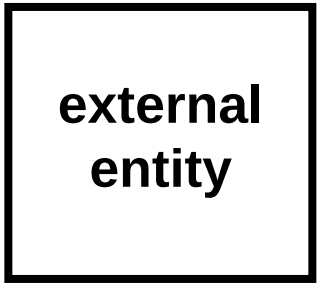
- Functional modelling is also called Process Modelling.

- In functional modelling, Data Flow Diagrams are used.
- There are various DFD notations such as:
  - Yourdon & Coad notation
  - Gane & Sarson notation

- For details of a process one of the followings can be used:
  - Flow Chart
  - Program Description Language (i.e. pseudocode)

Yourdon & Coad notation for DFD





Gane & Sarson notation for DFD

Double square

Source or destination of data

arrow

Flow of data

Rounded rectangle

Process that transforms a flow of data

Open-ended rectangle

Store of data

External Entity

external entity

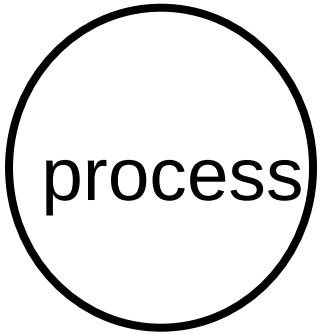
A producer or consumer of data

Examples: a person, a device, a sensor

Another example: computer-based system

Data must always originate somewhere and must always be sent to something

Process

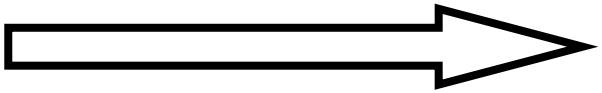


A data transformer (changes input to output)

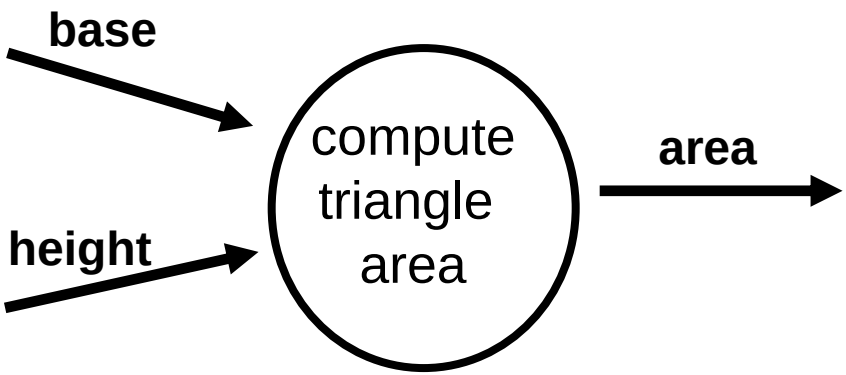
Examples: compute taxes, determine area, format report, display graph

Data must always be processed in some way to achieve system function

Data Flow



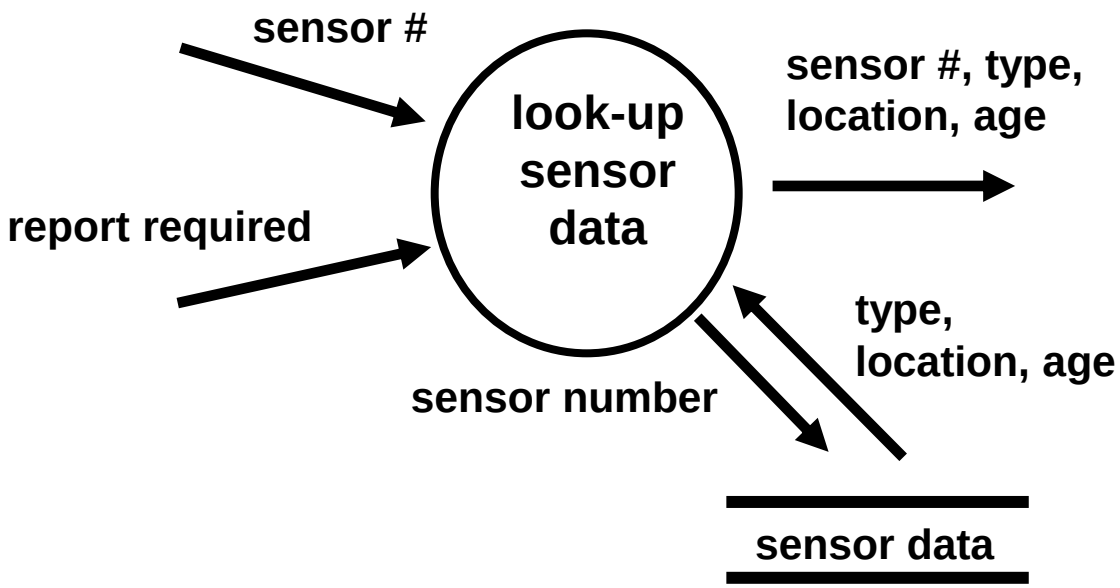
Data flows through a system, beginning as input and be transformed into output.



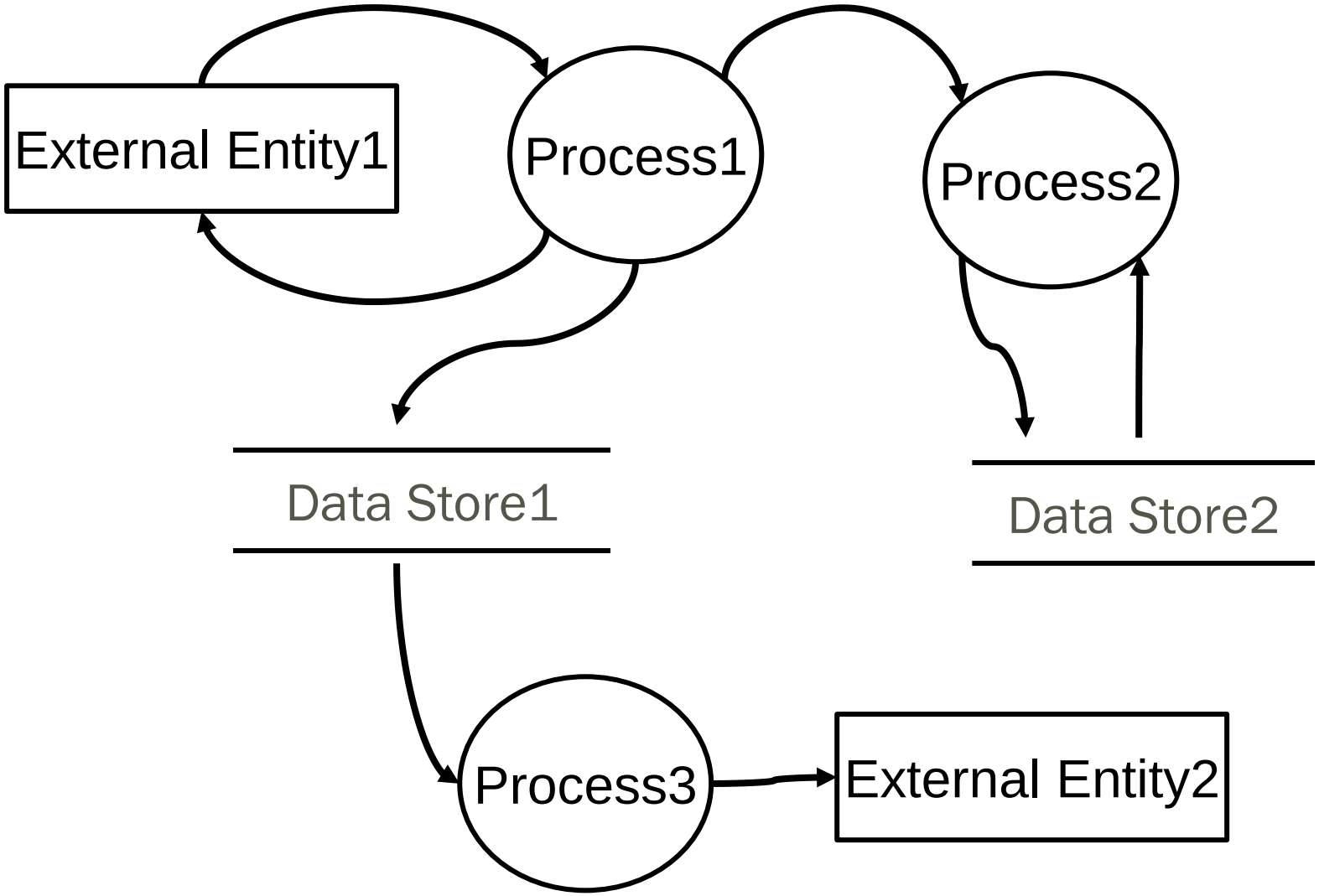
Data Stores

data store

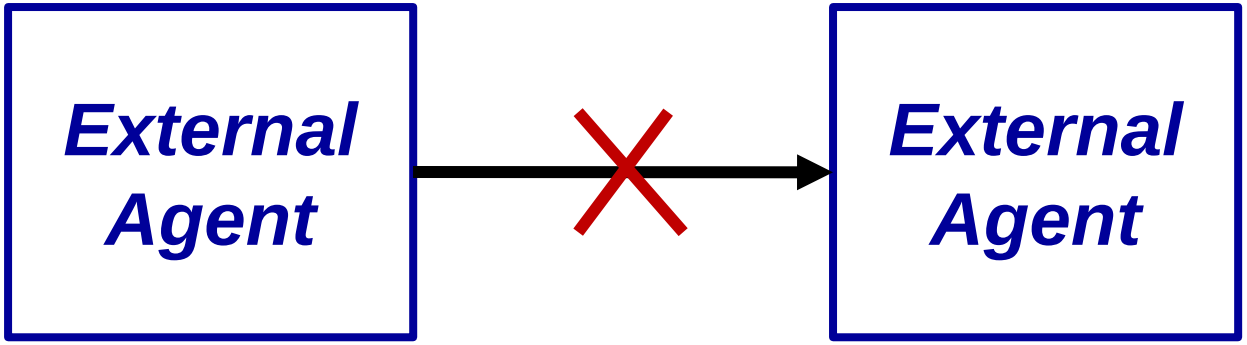
Data is often stored for later use.



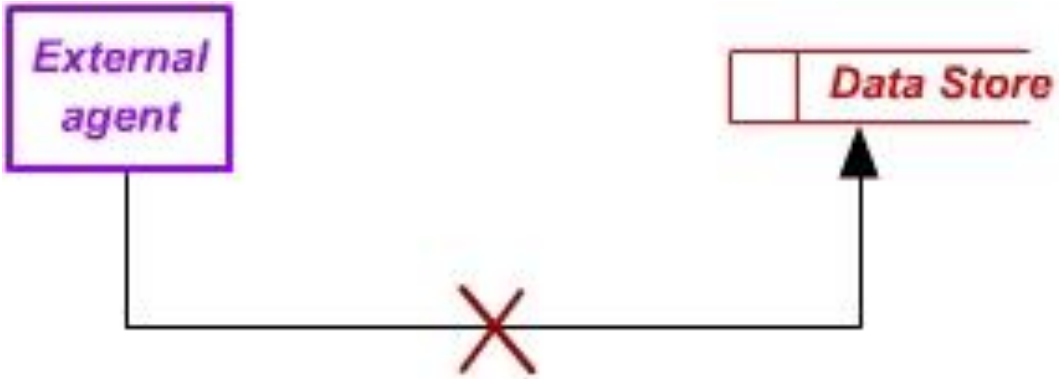
Example: Generic DFD

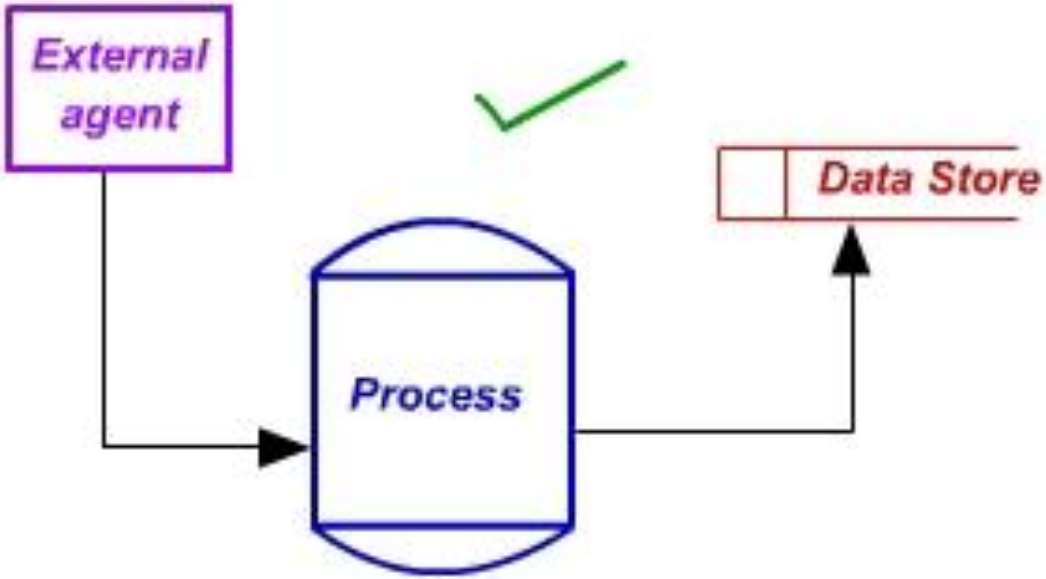


DFD Rules (1)

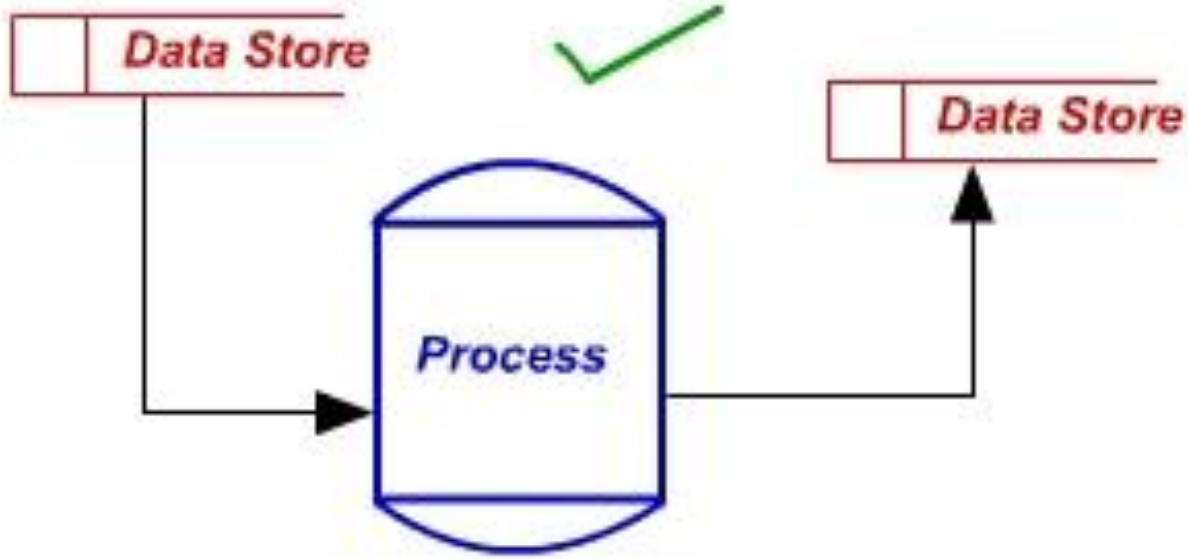


DFD Rules (2)

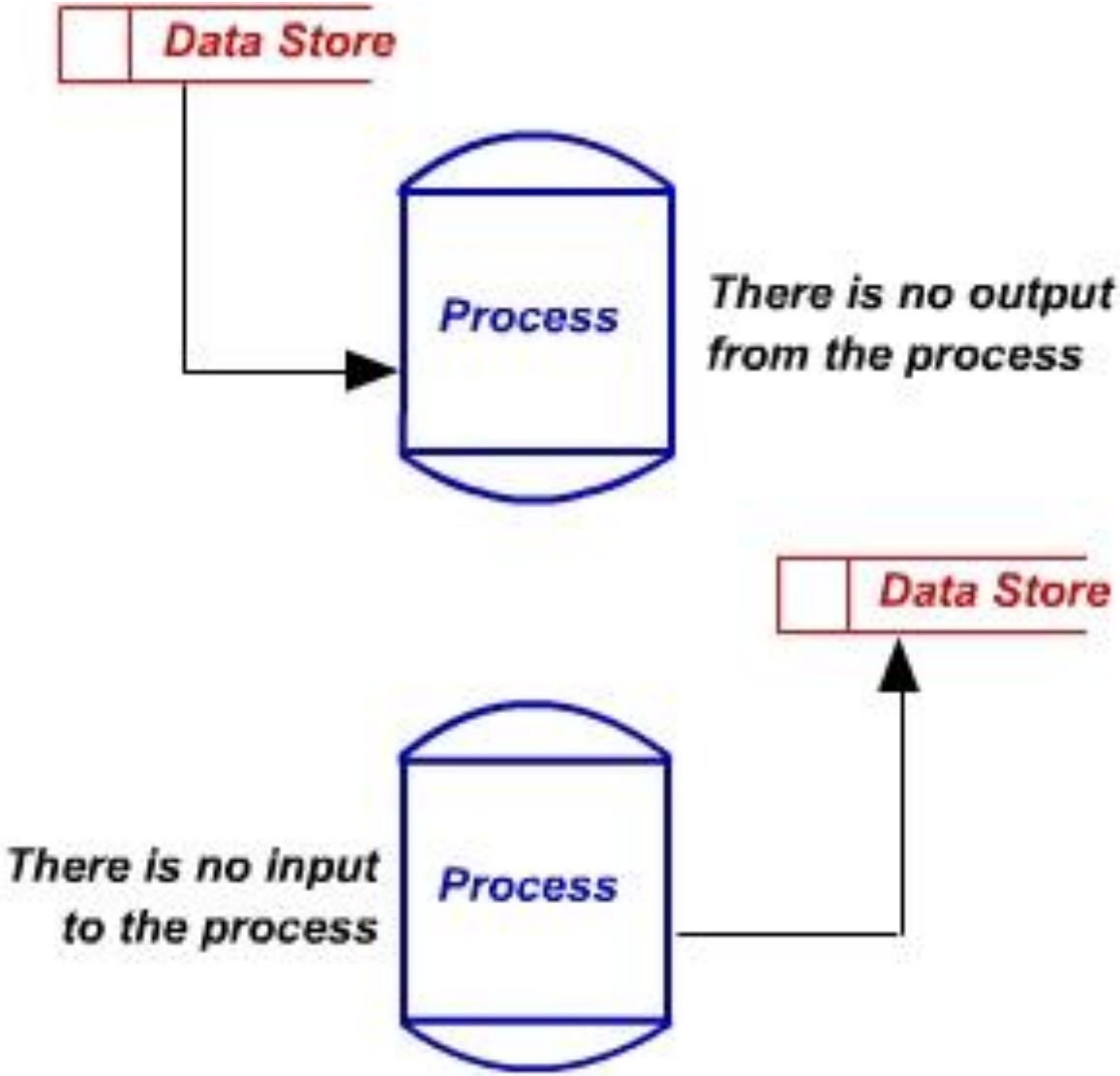

External agent
Data Store


External agent
Data Store
Process

DFD Rules (3)



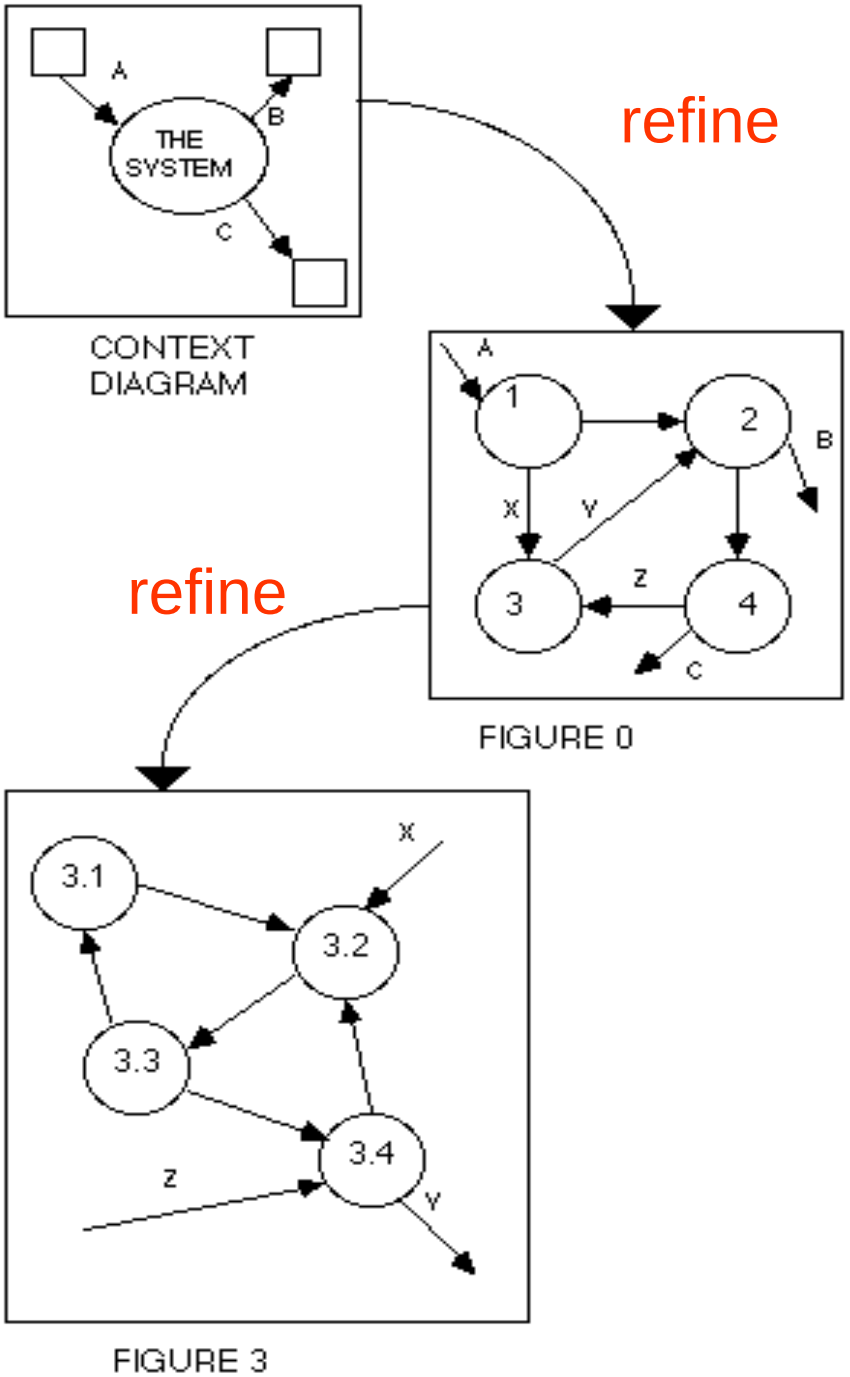
DFD Rules (4)


Data Store
Process
There is no output from the process
Data Store
There is no input to the process
Process

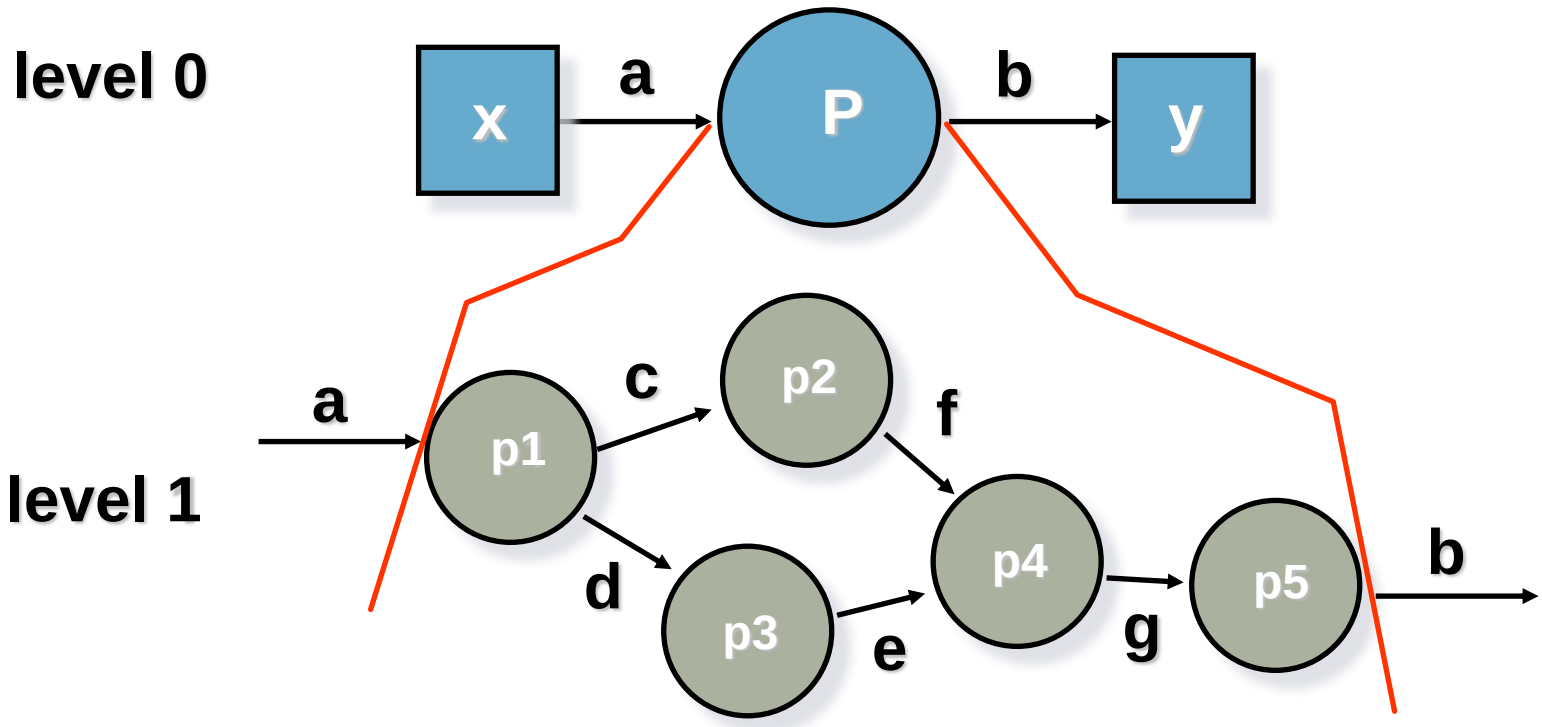
Data Flow Refinement

- DFD modelling is performed from level-0 to level-1, level-2, etc.
- A suggested expansion ratio between one level and the next level is 1:5
- Most systems require between 3 and 7 levels for an adequate flow model
- If a bubble does a number of different things, it needs further refinement.
- Each bubble is refined until it does just one thing
- The expansion ratio decreases as the number of levels increase

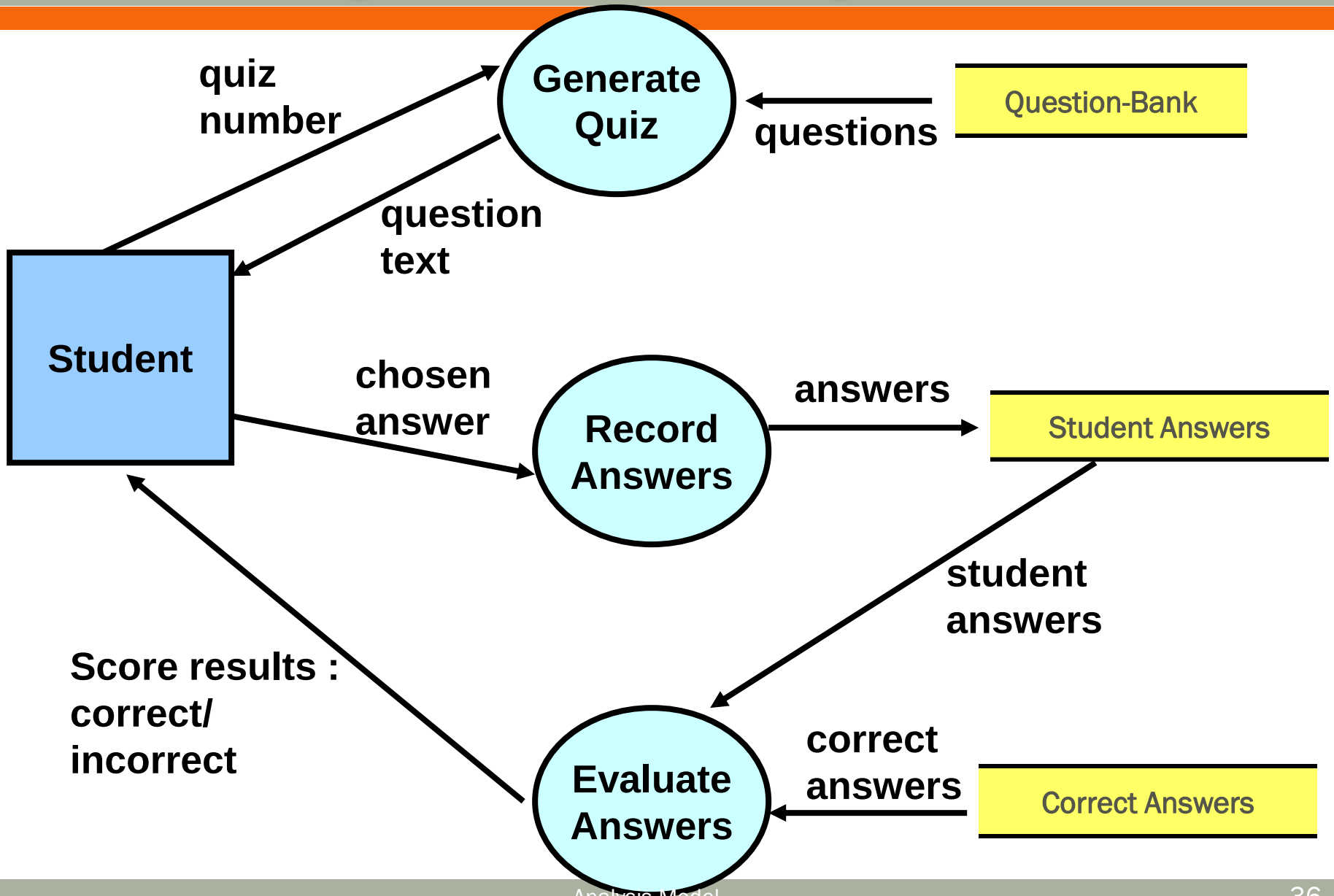
Example of Data Flow Refinement



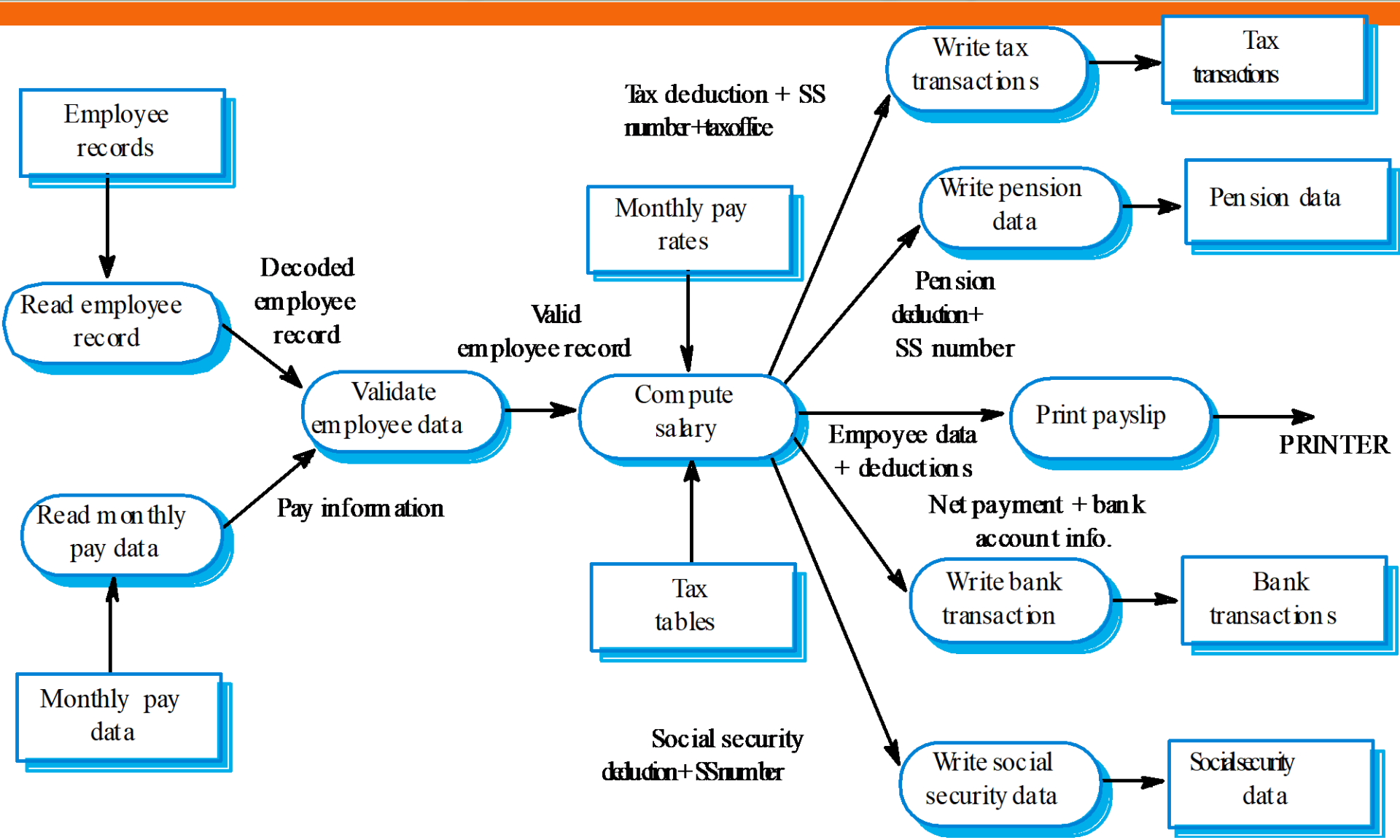
Example Data Flow Hierarchy



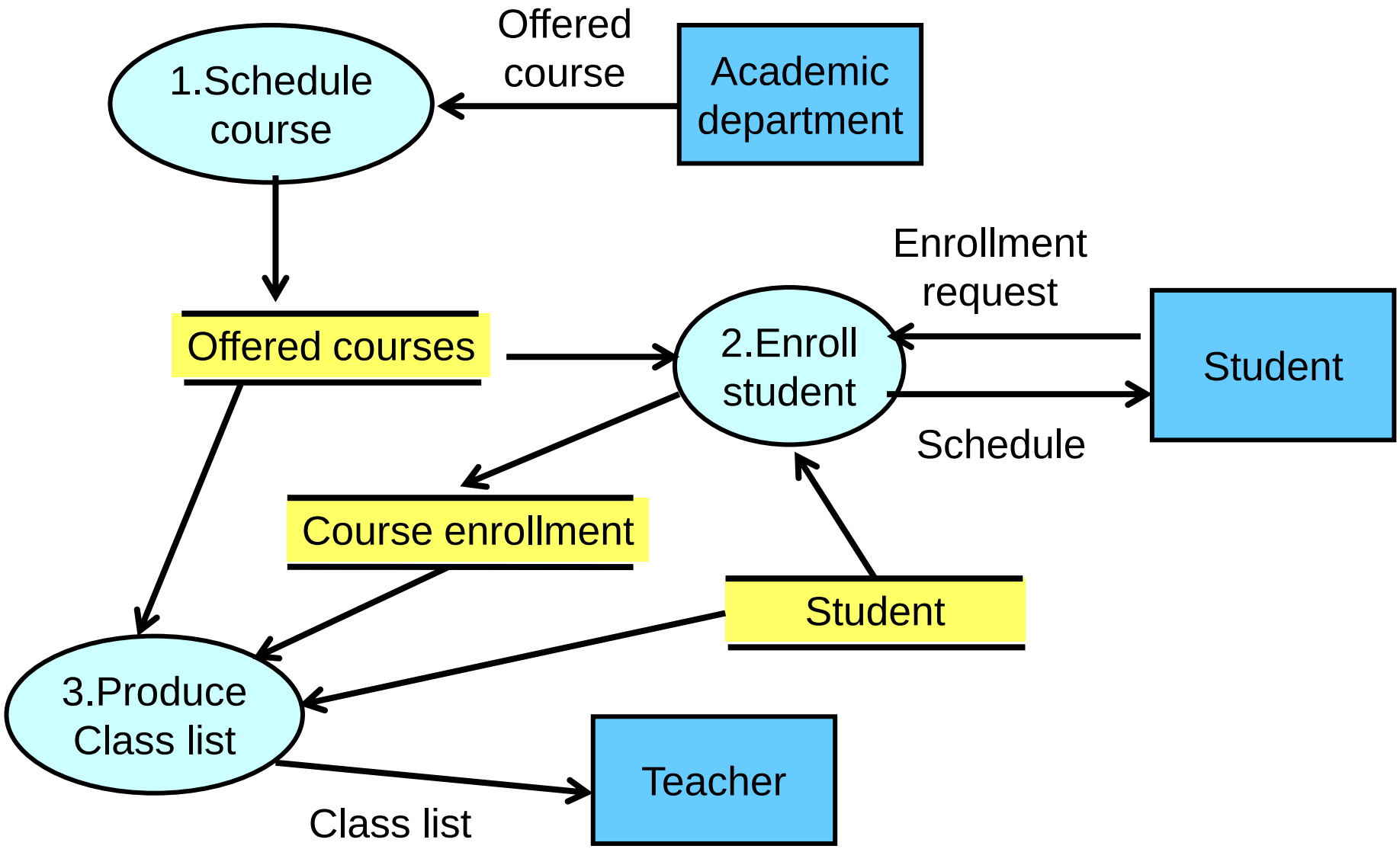
Example: DFD for Quizzing Software



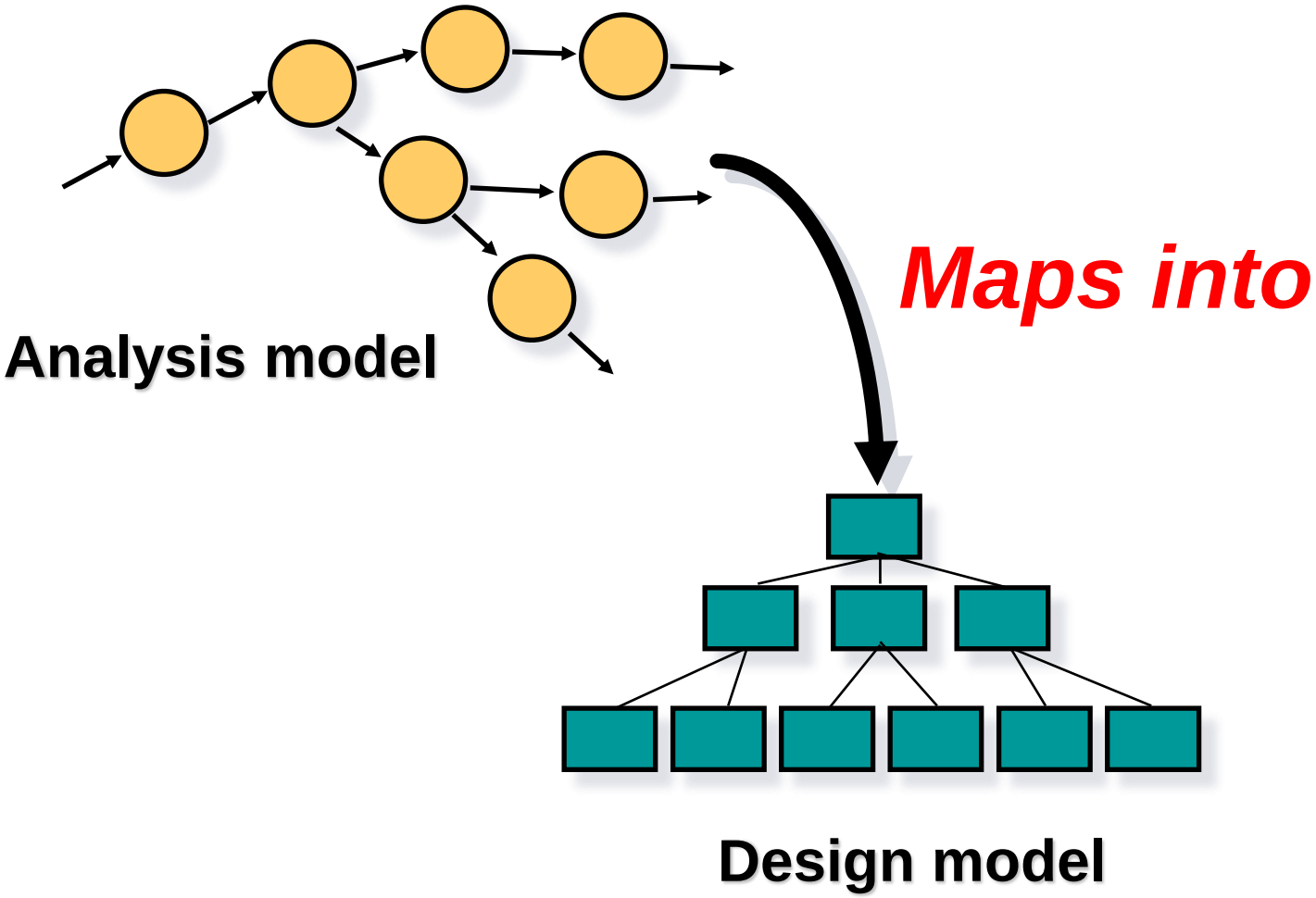
Example: DFD for Employees



Example: DFD for Courses



DFDs: A Look Ahead



1. Requirements Analysis
2. Structured Analysis
    1. Data Model
    2. Functional Model
    3. Behavioural Model

The Behavioural Model

6.2.3

The Behavioural Model

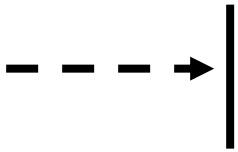
- In behavioural modelling, Control Flow Diagrams and State Transition Diagrams are used.
- Control Flow Diagrams is mostly used in Embedded or Real-time software development.
- The control flow diagram is superimposed on the DFD and shows events that control the processes noted in the DFD.
- Control flows (events and control items) are noted by dashed arrows.

Control Flow Diagrams

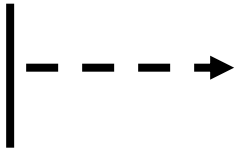
- Represents “events” and the processes that manage events
- An “event” is a Boolean condition that can be ascertained by:
  - listing all sensors that are "read" by the software.
  - listing all interrupt conditions.
  - listing all "switches" that are actuated by an operator.
  - listing all data conditions.
  - Examining the processing narrative, review all "control items" as possible CSPEC inputs/outputs.
  - A CSPEC is shown with a **State Transition Diagram**.

Control Flow Diagrams

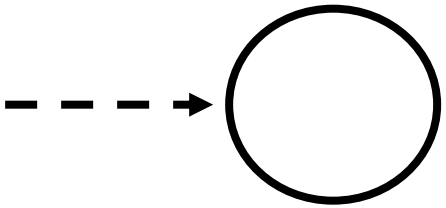
- a dashed arrow entering a vertical bar is an input



- a dashed arrow leaving a process implies a data condition

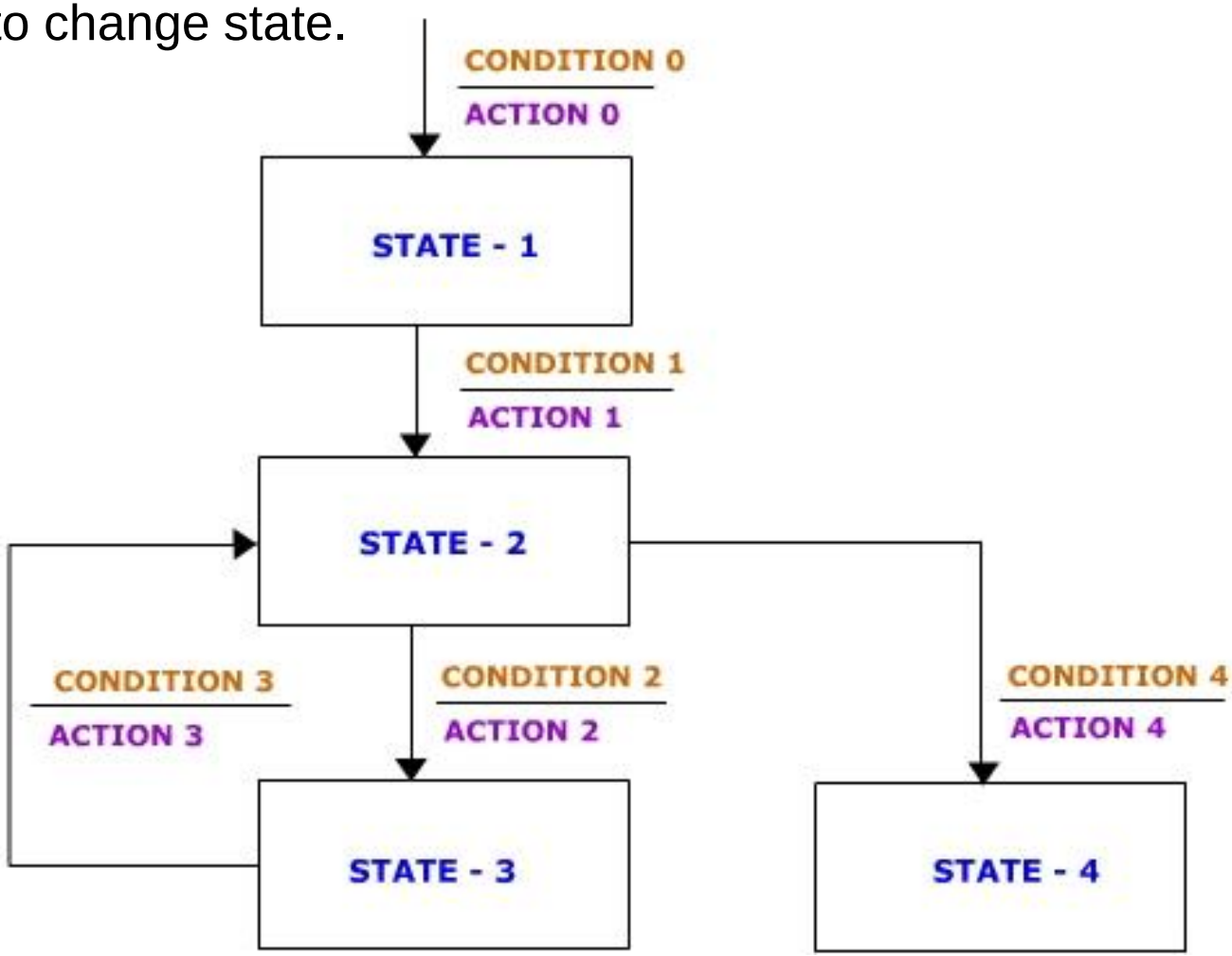


- a dashed arrow entering a process implies a control input read directly by the process



State Transition Diagrams

- STD can be used to model the state changes of the system.
- A system is in a state and will remain in that state till a condition and an action force it to change state.



Example:

Vending Machines
Management Software

Statement of Software Scope (1)

- You've been asked to develop a management software for a company which maintains a large number of vending machines (self-service machines to sell snack foods).
- Vending machines are at several locations across the city.
- Each location can have one or more machines.
- Vending machines need to be refilled with different quantities depending on the consumption at each location.

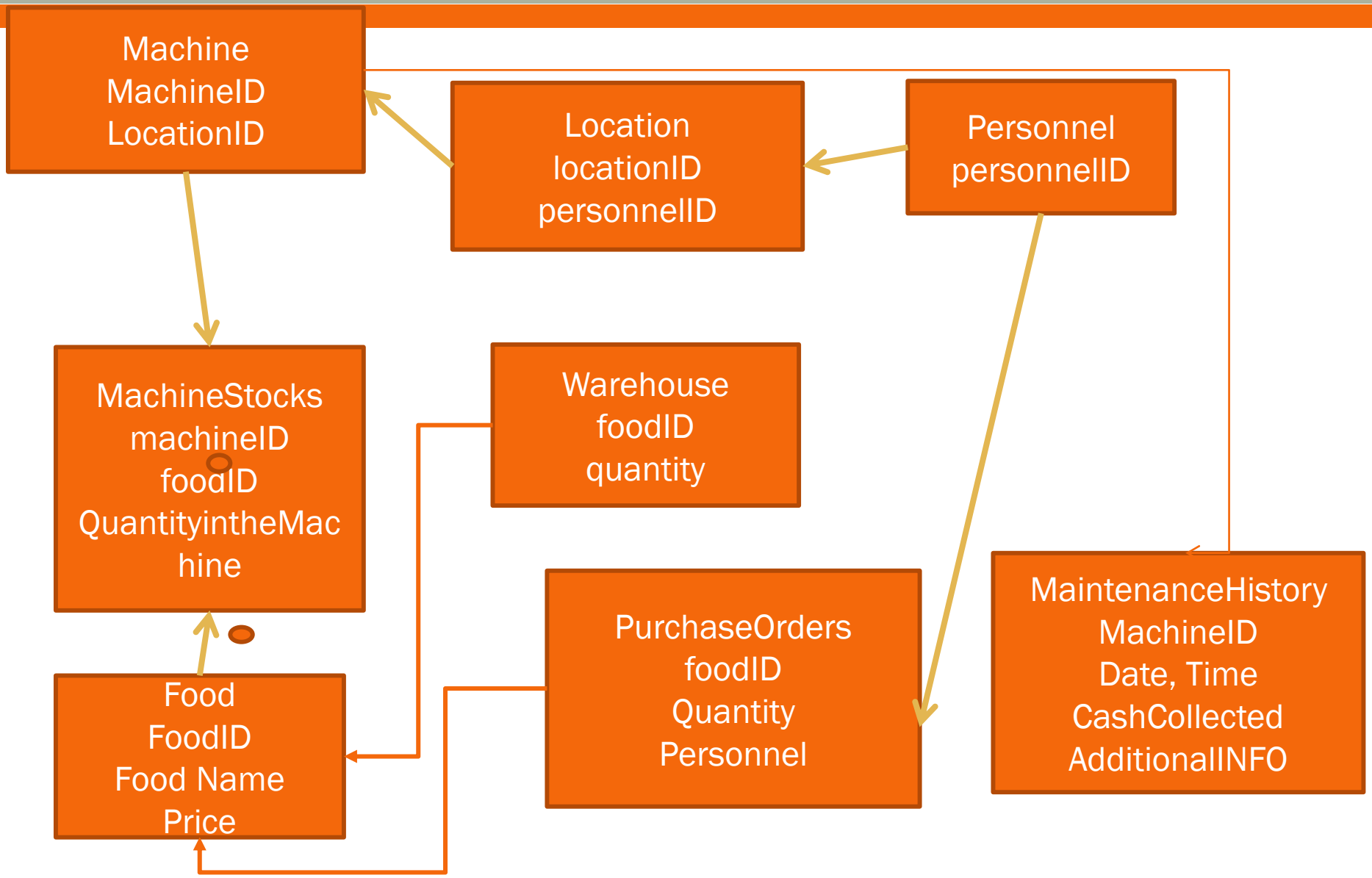
Statement of Software Scope (2)

- Each location is served by one service personnel.
- All foods are stored in the company's warehouse.
- Before a personnel leaves for servicing, he requests foods from the warehouse for refilling.
- After returning from servicing, the service personnel submits the cash he collected from each machine to the company; returns any unused foods; and informs the company of any problems with the machines.
- When the food stock gets low, a purchase order is generated.

Statement of Software Scope (3)

- The company wants to manage their business using the software that keeps track of the
  - locations,
  - machines,
  - service personnel,
  - food stocks,
  - maintenance history for machines,
  - the amount of food requested and returned by each personnel,
  - total cash generated per machine, per location,
  - details of any purchase orders generated.

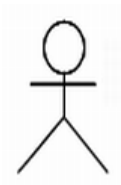
- Daily reports (such as total cash report, maintenance summary report, purchase order) will need to be generated.


Machine
MachineID
LocationID
Location
locationID
personnelID
Personnel
personnelID
MachineStocks
machineID
foodID
QuantityintheMac
hine
Warehouse
foodID
quantity
PurchaseOrders
foodID
Quantity
Personnel
MaintenanceHistory
MachineID
Date, Time
CashCollected
AdditionalINFO
Food
FoodID
Food Name
Price

System Outline

Location 1

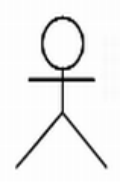
- ❑machine 1
- ❑machine 2



Service person 1

Location 2

- ❑machine 1
- ❑machine 2
- ❑machine 3



Service person 2

Warehouse

- ❑food 1
- ❑food 2
- ❑food 3
- ❑ . . .

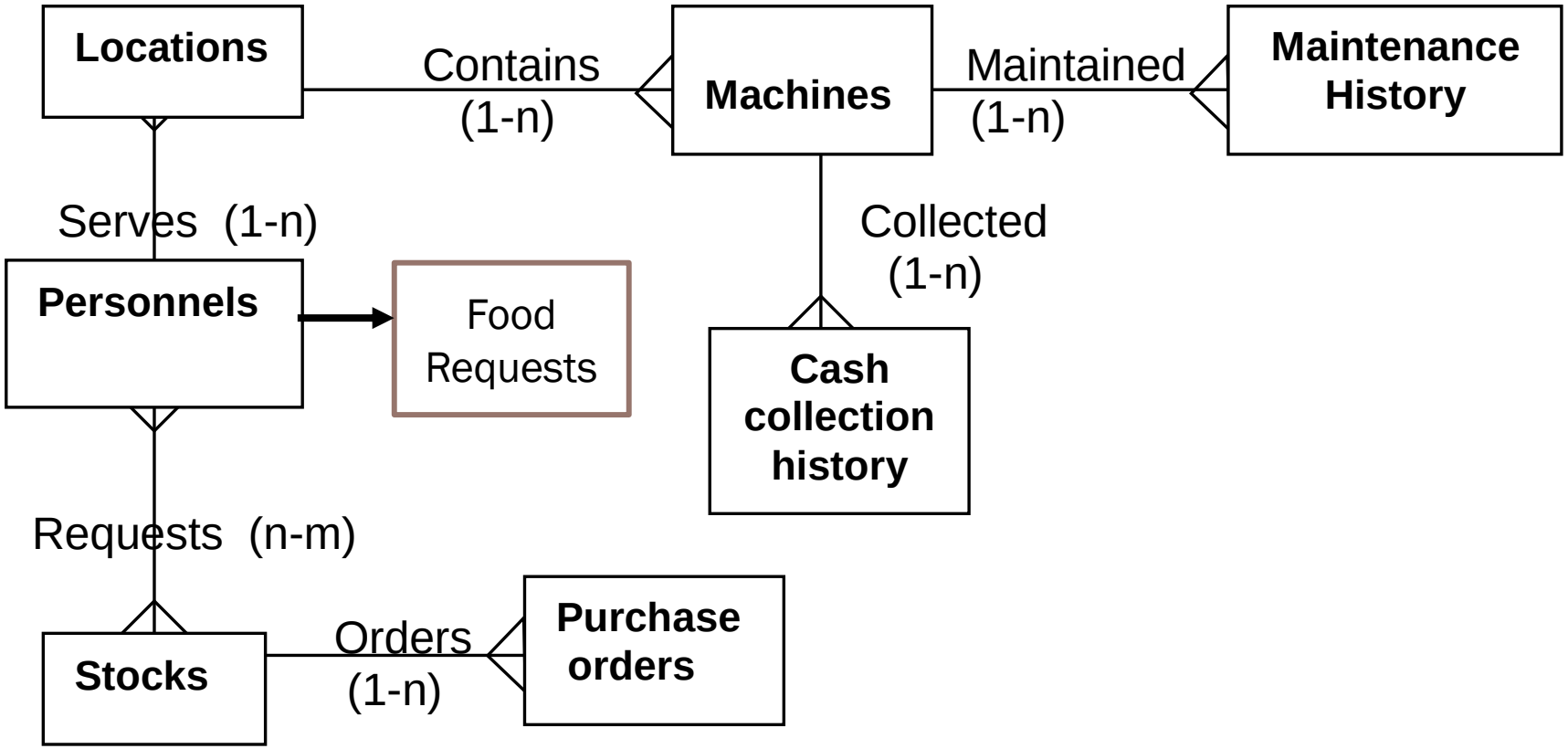
Tasks

- Draw an Entity Relationship Diagram that describes the relationships between the different data entities.
  - For each relationship, name the relationship and define its cardinality (1-1, 1-n, or n-m).
  - For each entity, list all data items.

- Produce Level-0 and Level-1 Data Flow Diagrams that captures the main processes, data flows, information sources and data stores of this application.

- Produce a Program Structure Chart.

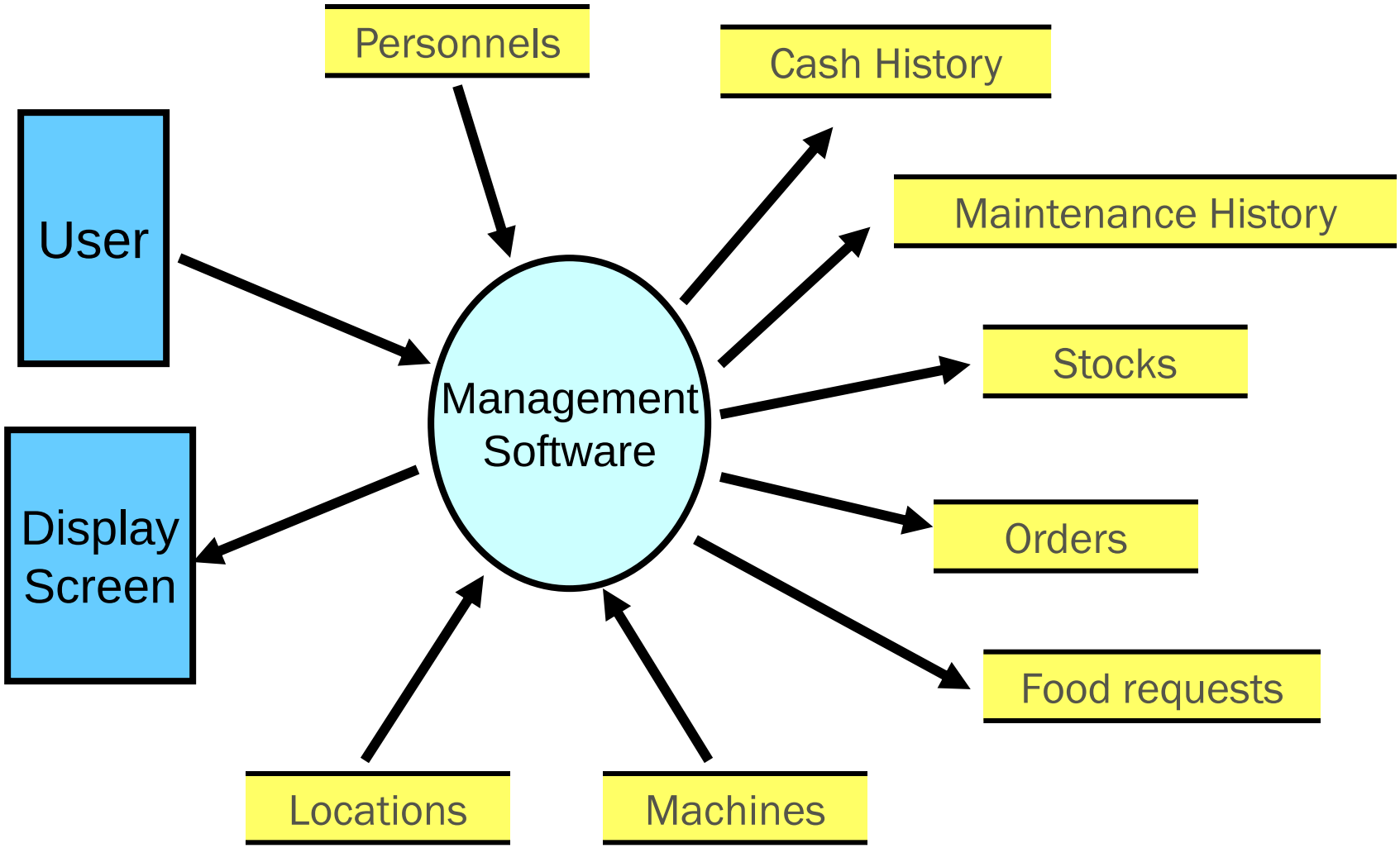
Entity Relationship Diagram (ERD)



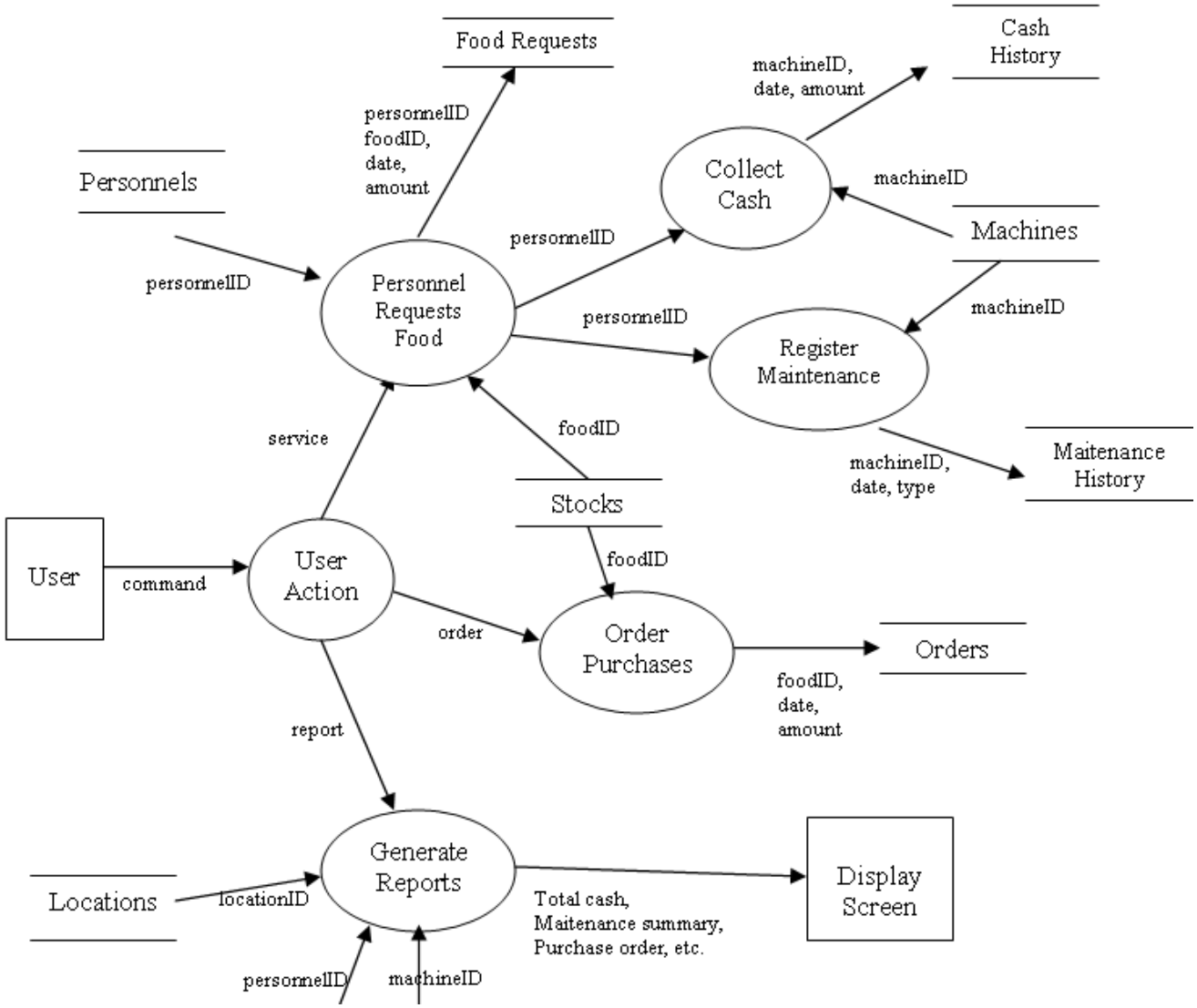
Entities

ENTITIY	DATA ITEMS
Locations	Location_ID, Address, Number of consumers, ServicePersonnel_ID
Machines	Machine_ID, Location_ID, Frequency of refilling
Personnels	Personnel_ID, Personnel name
Stocks	Food_ID, Food name, Current amount
Food_Requests	Personnel_ID, Food_ID, Date of request, Amount of request, Returned amount
Cash_Collection_History	Machine_ID, Date of collection, Amount of cash
Maitenance_History	Machine_ID, Date of maintenance, Type of maintenance
Purchase_Orders	Order_ID, Food_ID, Date of order, Amount of order

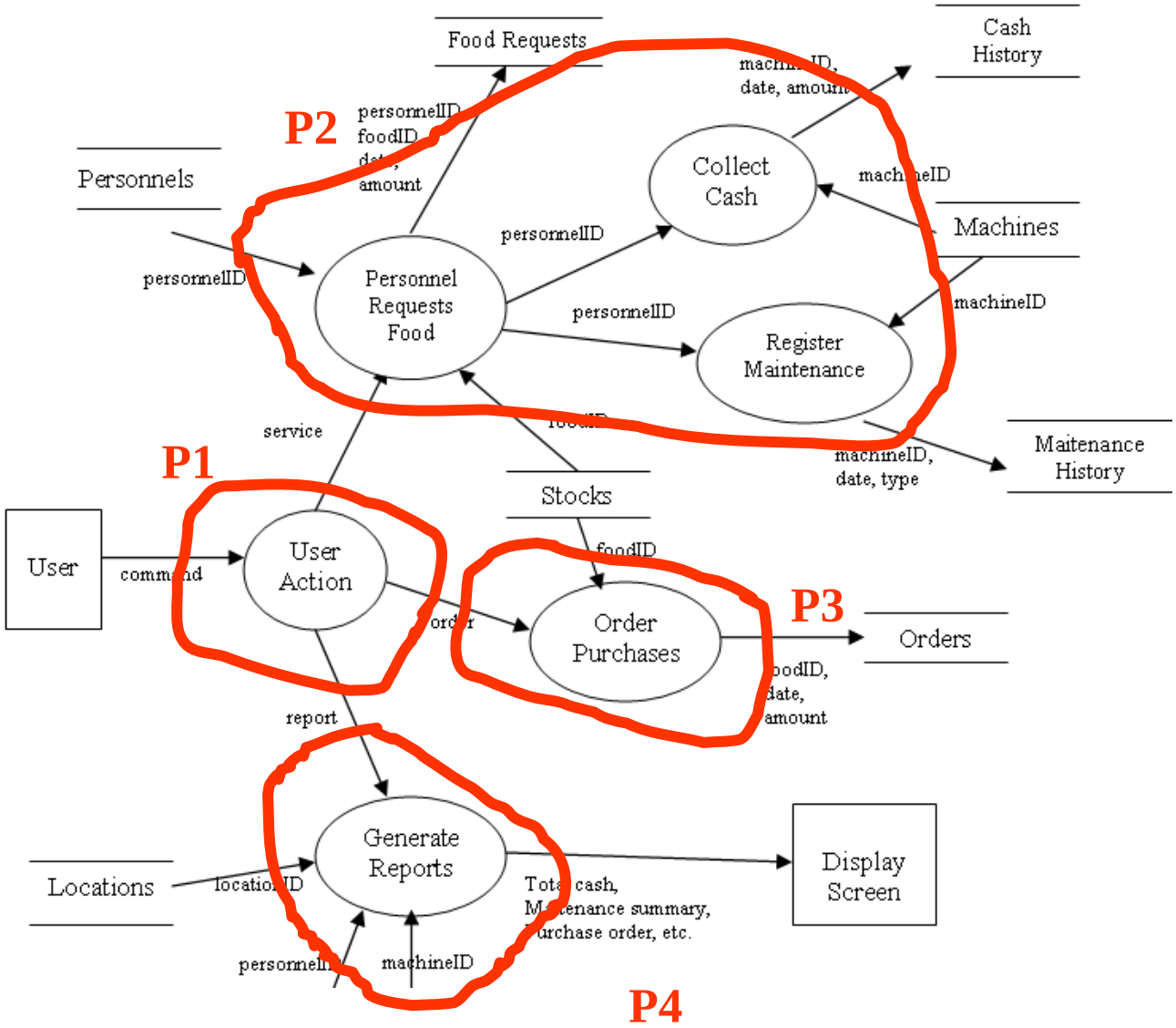
Level-0 DFD


Personnels
Cash History
User
Maintenance History
Management Software
Stocks
Display Screen
Orders
Food requests
Locations
Machines

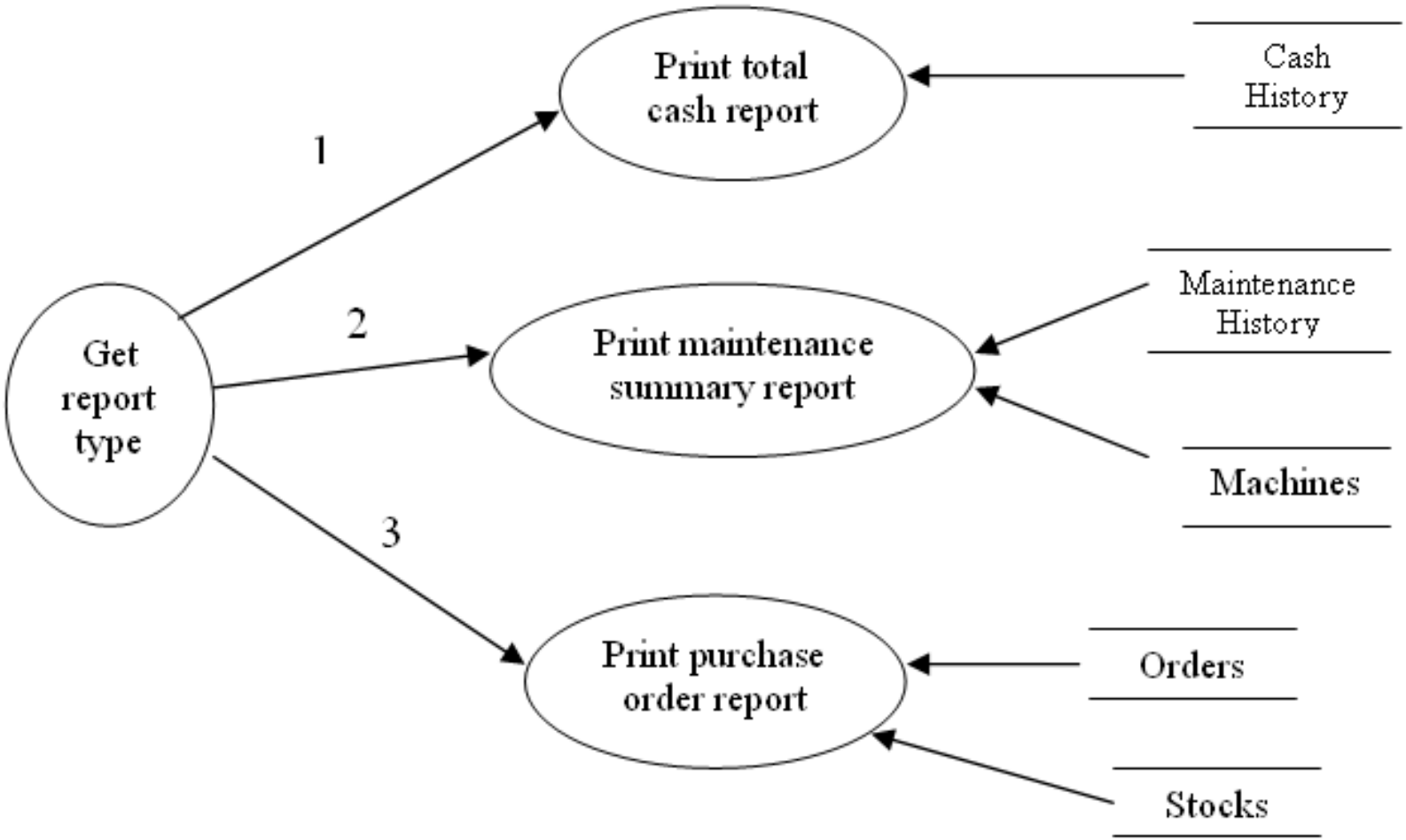
Level-1 DFD


Food Requests
Cash History
machineID, date, amount
personnelID foodID, date, amount
Personnels
Collect Cash
machineID
Machines
personnelID
personnelID
Personnel Requests Food
personnelID
machineID
Register Maintenance
service
foodID
machineID, date, type
Maitenance History
Stocks
User
command
User Action
foodID
order
Order Purchases
Orders
foodID, date, amount
report
Generate Reports
Locations
locationID
Display Screen
Total cash, Maitenance summary, Purchase order, etc.
personnelID
machineID

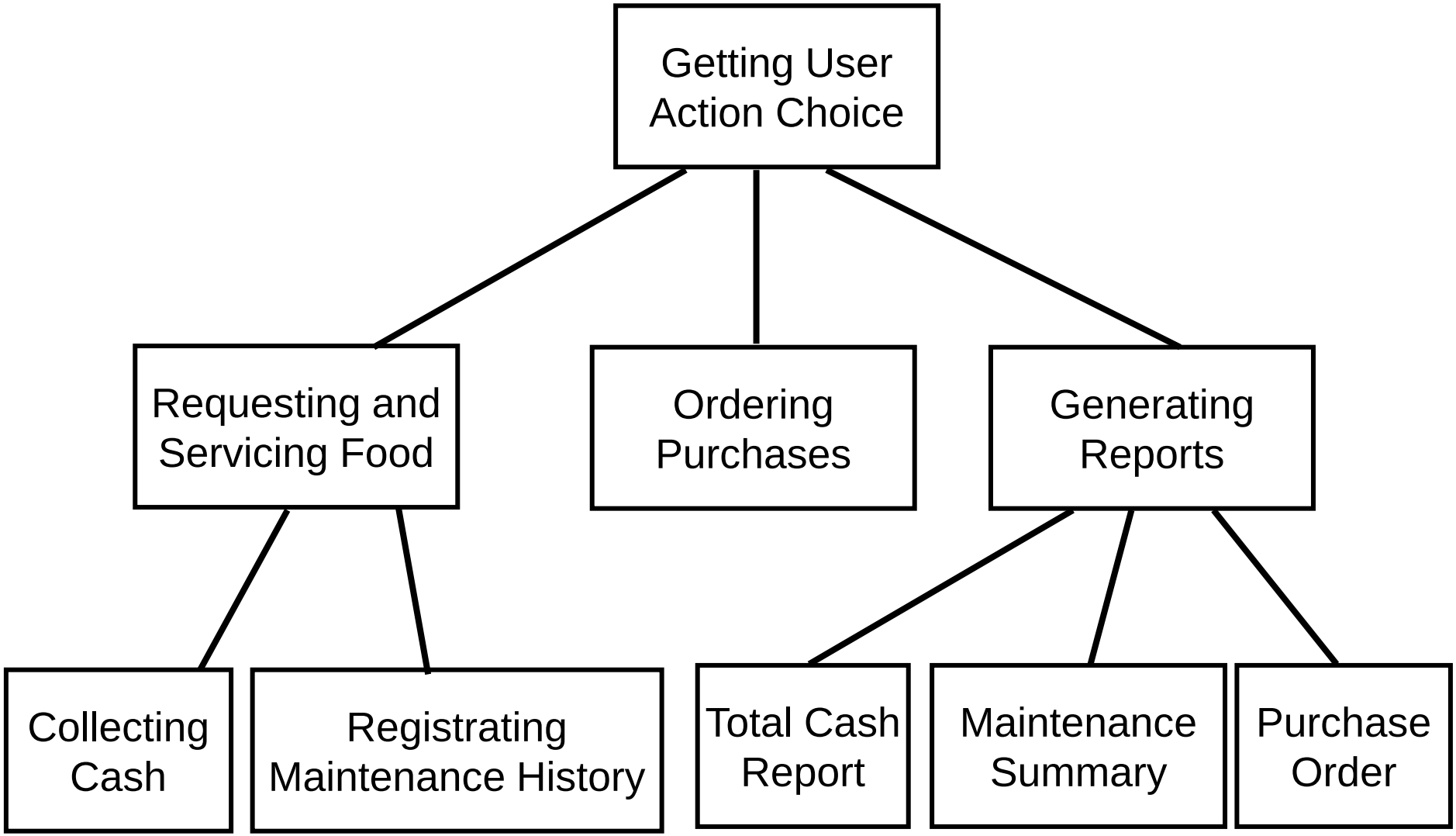
Isolating the flows in Level-1 DFD



Level-2 DFD for P4



Program Structure Chart



Example:

Technical Service
Management Software

Statement of Software Scope (1)

- A technical service firm needs a web-based software to keep track of maintenance and repairment operations for their customers' devices such as combi, air conditioner, laundry machine, refrigerator, etc.

- <u>The followings are functional requirements:</u>

- A "Service Request Form" must be filled for all kinds of service requests. The form must contain fields for *customer name, address, telephone, and a description of service* being requested.

Statement of Software Scope (2)

- The request will be tracked by a status code:
  - "Device will be picked up from customer"
  - "Device will be serviced at customer's place"
  - "Device is in service"
  - "Device waiting for delivery to customer"
  - "Delivery completed"

- A request can be done directly by a customer over the Internet, or an authorized personnel can record the request for the customer.

- Customer should be able to query the status of his service request.

Statement of Software Scope (3)

- The manager will assign a service request task to an available technician.

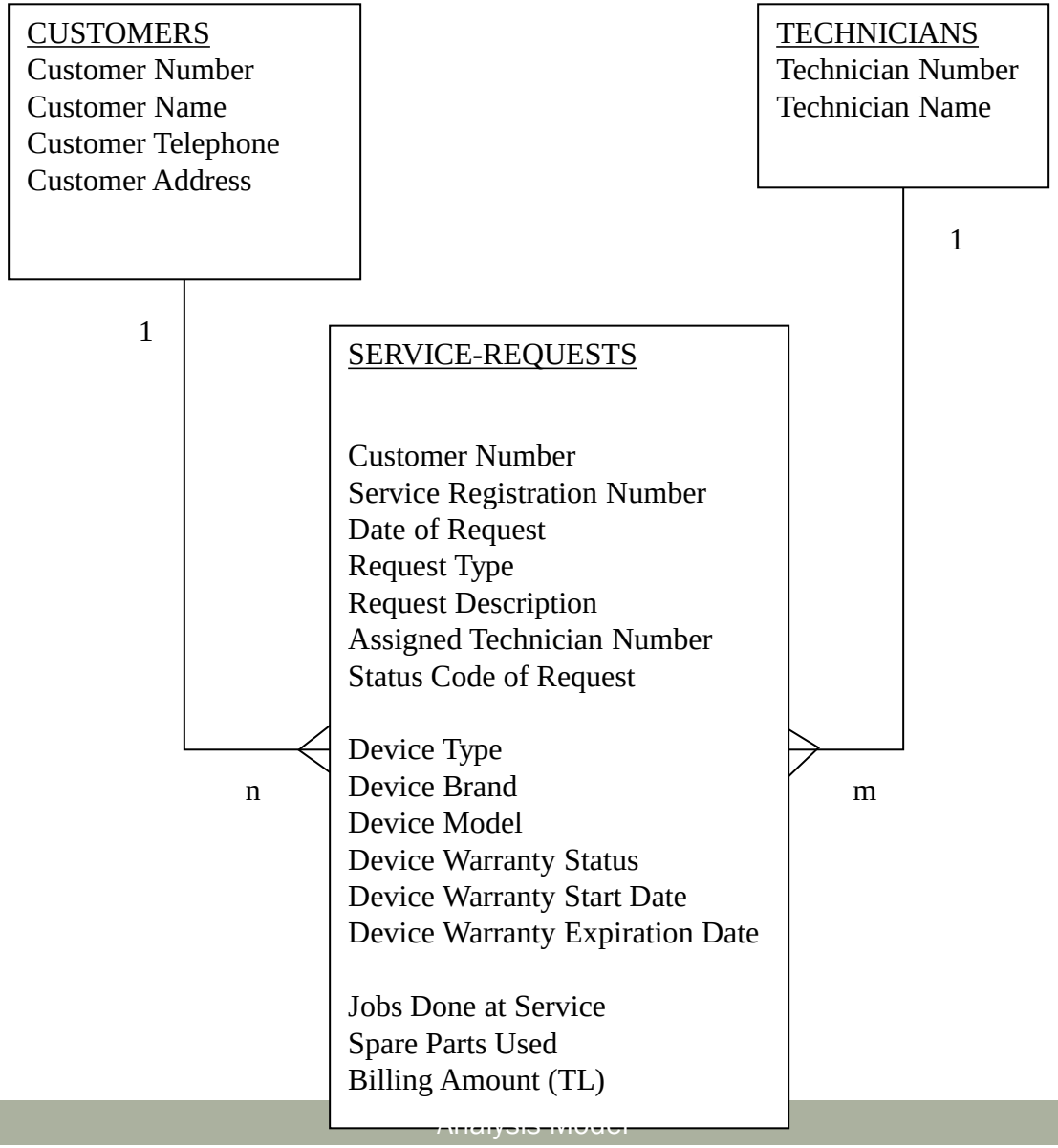
- For each service request the followings should be recorded: *Device information (device type, brand, model, warranty status, start date, expiration date); Jobs done at service, Spare parts used if any, Billing amount (TL).*

- For customers who has warranty agreement, periodic maintainances will be tracked. For this purpose, a list of devices which are sorted by warranty expiration date should be available.

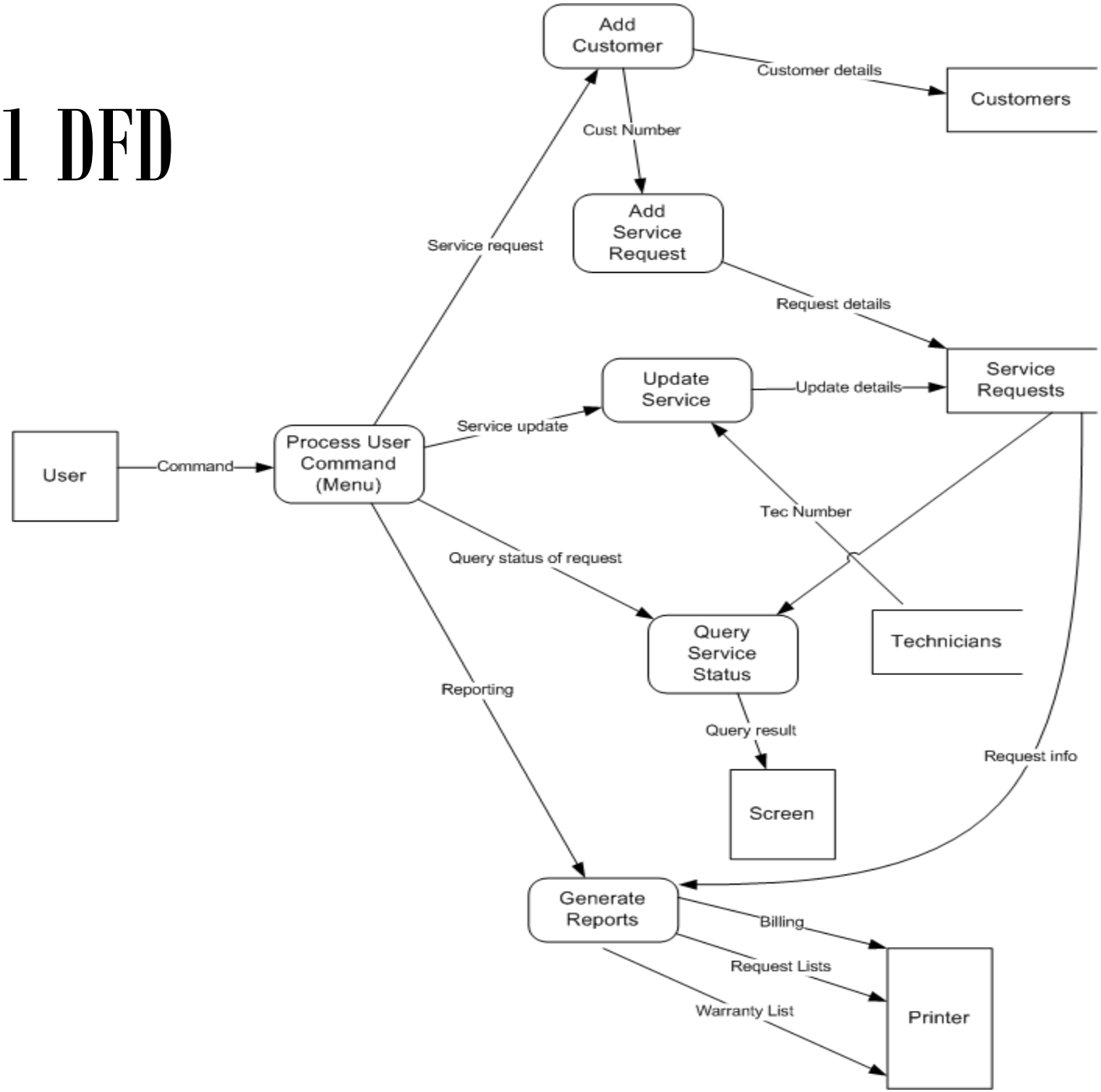
Statement of Software Scope (4)

- "Service Request Lists" should be available with different criteria:
  - by service registration number
  - by customer name
  - by status code
  - by device type
  - by request type
  - by date of request
  - by technician name

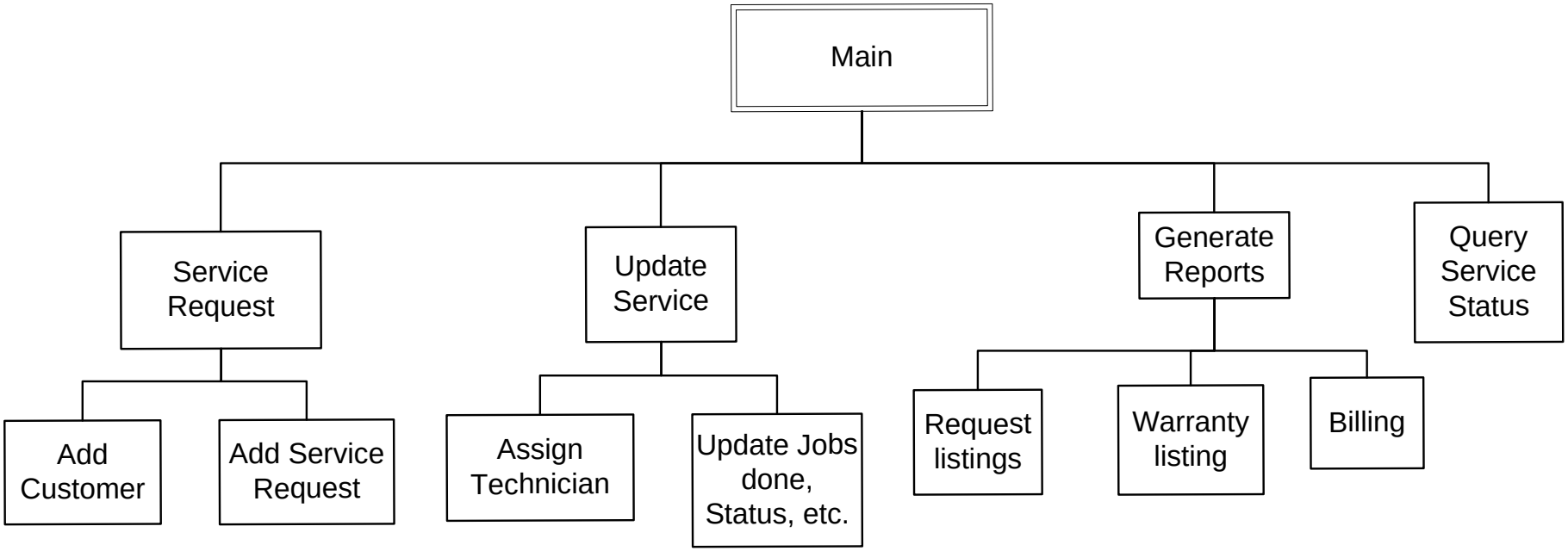
Entity Relationship Diagram (ERD)



Level-1 DFD


Add Customer
Customer details
Customers
Cust Number
Add Service Request
Service request
Request details
Service Requests
Update Service
Update details
Service update
User
Command
Process User Command (Menu)
Tec Number
Query status of request
Query Service Status
Technicians
Reporting
Query result
Request info
Screen
Generate Reports
Billing
Request Lists
Warranty List
Printer

Program Structure Chart



Wrap-up

This week we present

- Structural Analysis: Where the main focus of the analysis stage is handling static and dynamic system behaviour separately
- Object Oriented Analysis: Where the main focus of the analysis is to represent the objects inherent in the requirements as classes with specific data and behaviour

Next Week

- We will be covering *Architectural Models and Model Driven Engineering*!!!