



**SOFTWARE ENGINEERING**

**Week 8**

**Analysis Model**

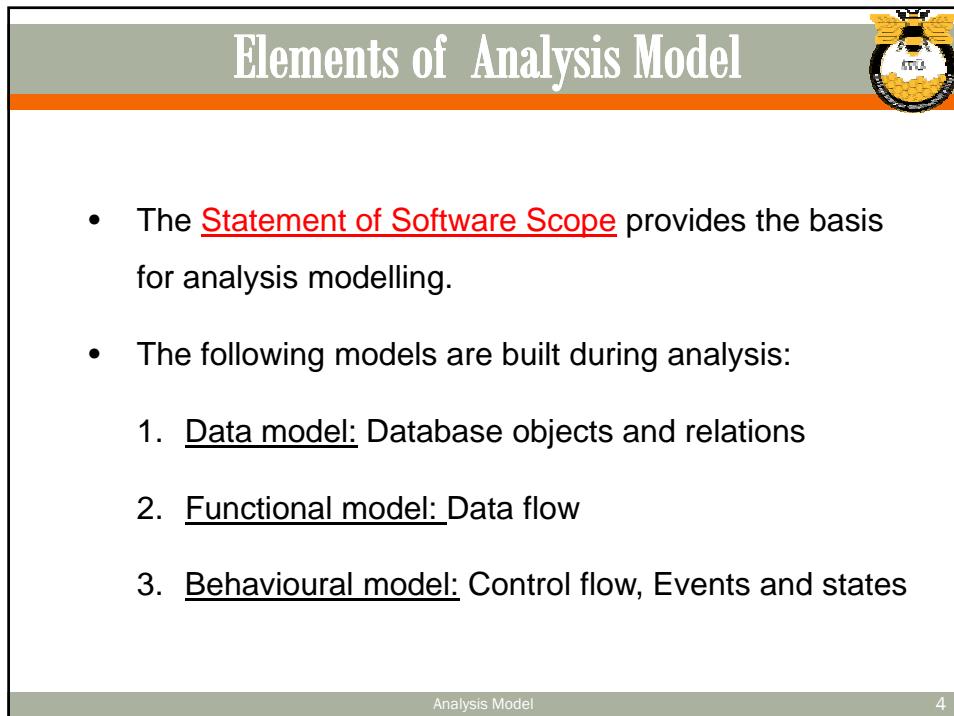
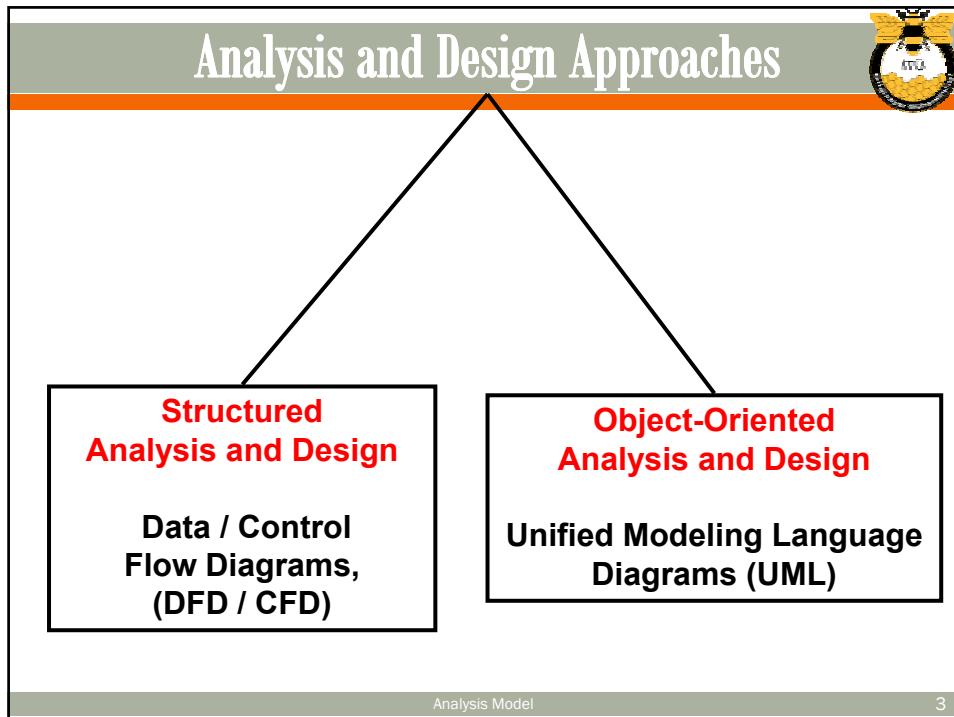
## Agenda



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

Analysis Model

2



## Modelling the Data Domain



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

Analysis Model

5

## 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

Analysis Model

6

# 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

Analysis Model

7

1. Data Model ←
2. Functional Model
3. Behavioural Model

## The Data Model

» 8.1 »

Analysis

## 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)





Analysis Model 9

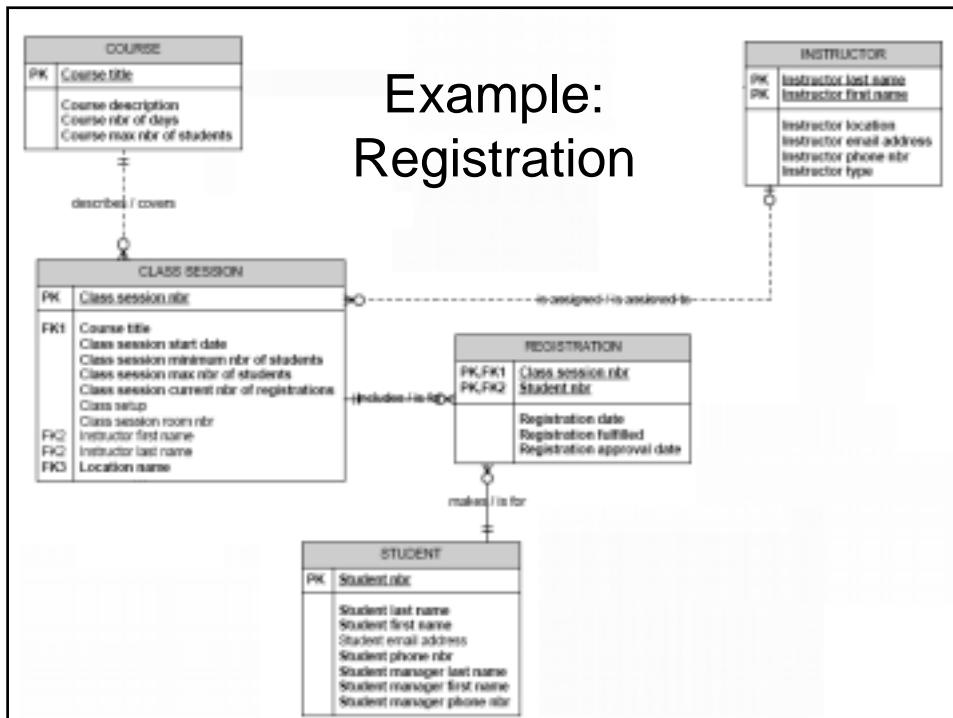
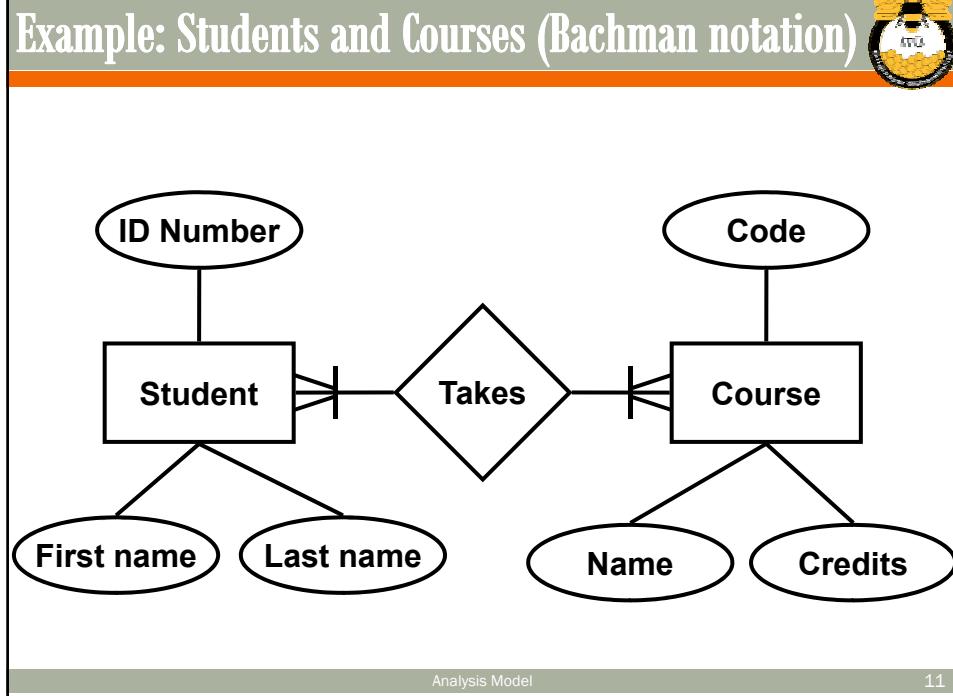
## Relationship Symbols

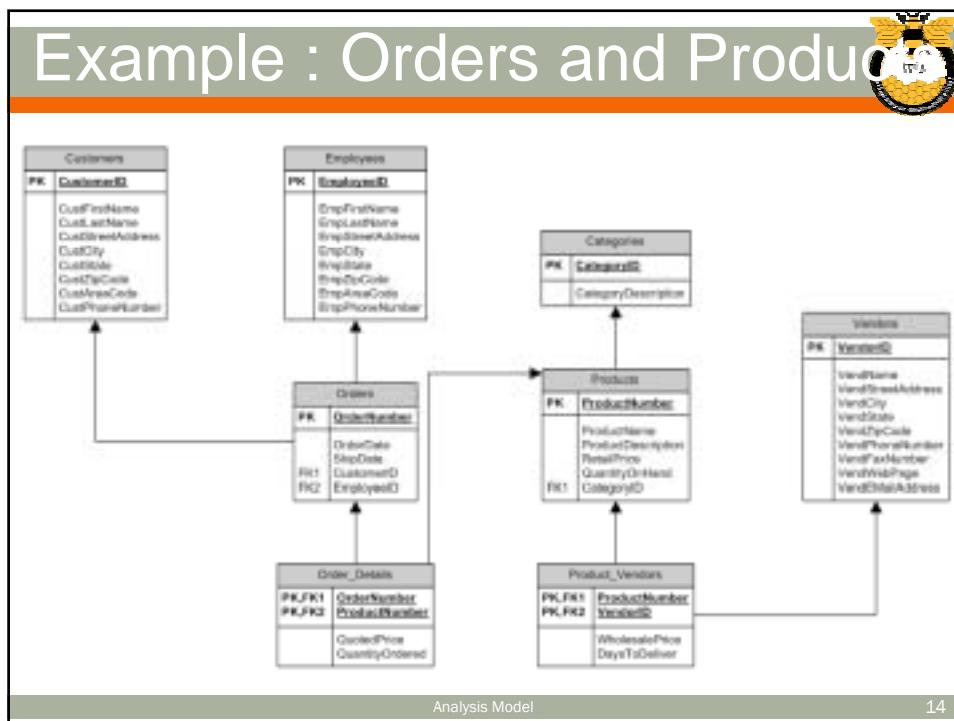
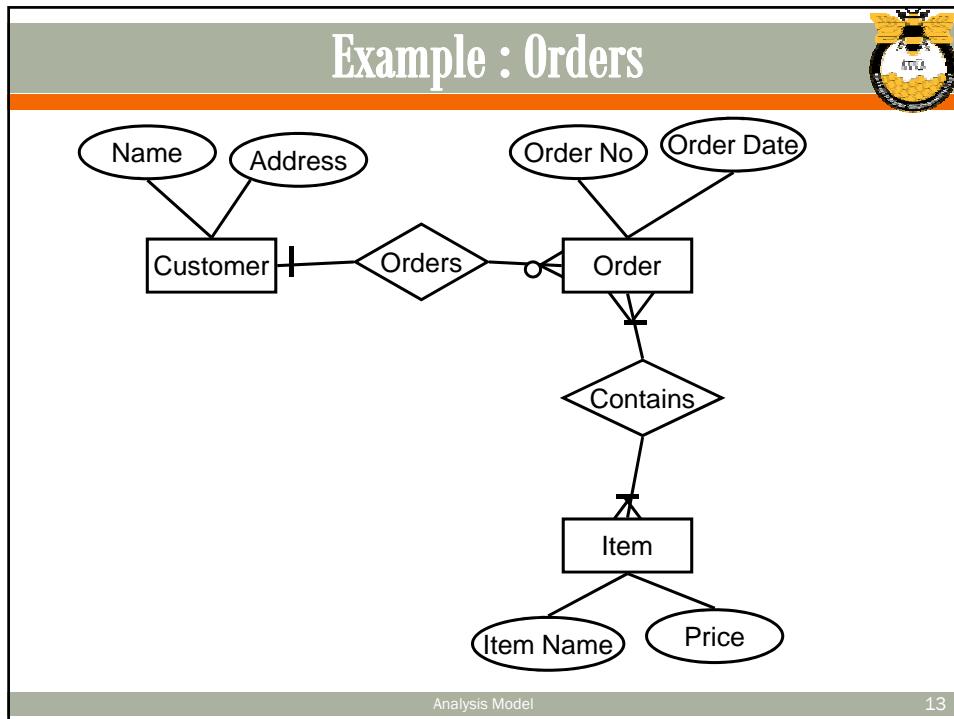


A ————— B	One to one, mandatory
A ————— O B	One to one, optional
A —————> B	One to many, mandatory
A ————— O ————— B	One to many, optional

**Cardinality**      **Modality**

Analysis Model 10





# Data Dictionary



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

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

Analysis Model

15

# Data Dictionary Example



<u>name:</u> <u>aliases:</u> <u>where used/how used:</u>	<u>telephone number</u> none <u>access against set-up (output)</u> <u>dial phone (input)</u>
<u>description:</u> 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 *	

Analysis Model

16

## Data Dictionary Example



**iTÜ Student ID = 04 – 009 - 0761**

```
graph TD; A[Electrical-Electronics faculty] --> B[Entrance year]; C[Order number]
```

Analysis Model      17

1. Data Model
2. Functional Model ←
3. Behavioural Model

## The Functional Model

8.2

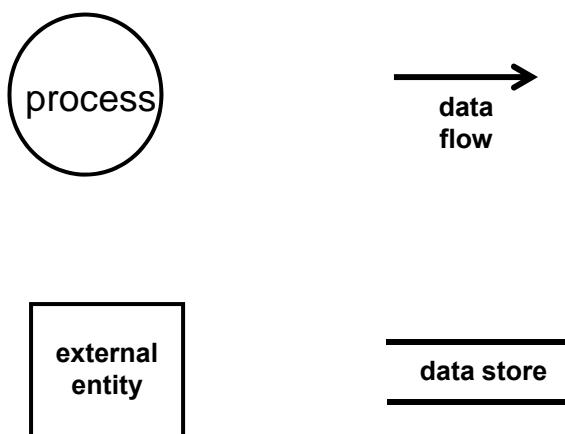
Analysis

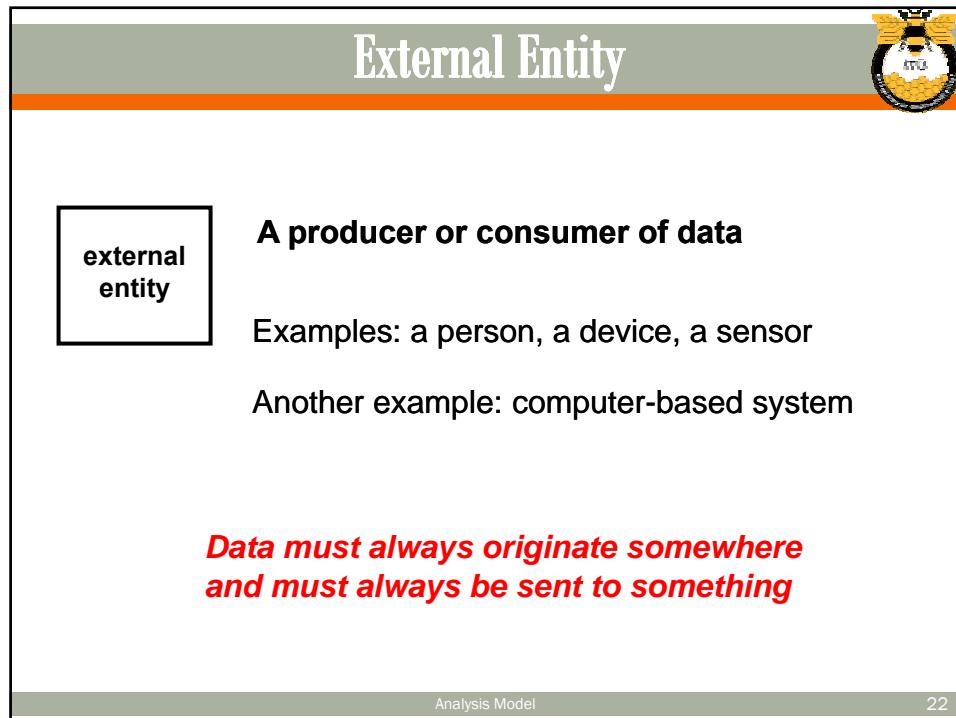
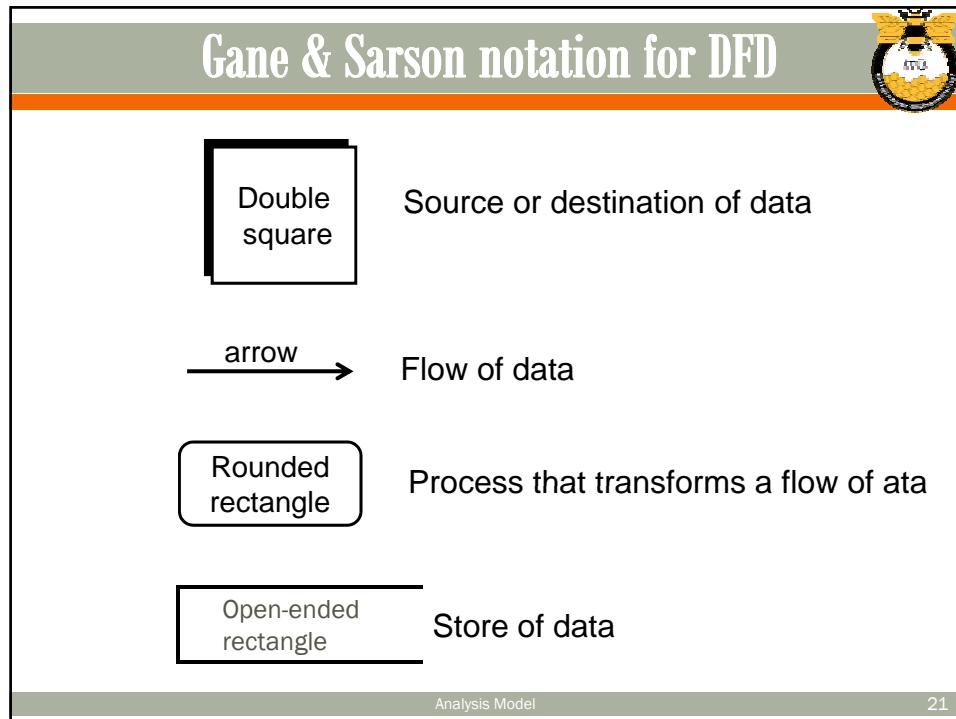
## The Functional Model



- 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





## Process



process

**A data transformer (changes input to output)**

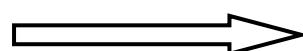
Examples: compute taxes, determine area, format report, display graph

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

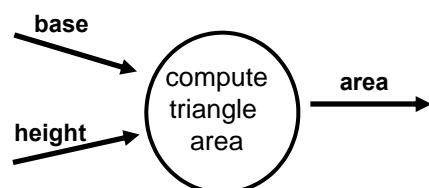
Analysis Model

23

## Data Flow

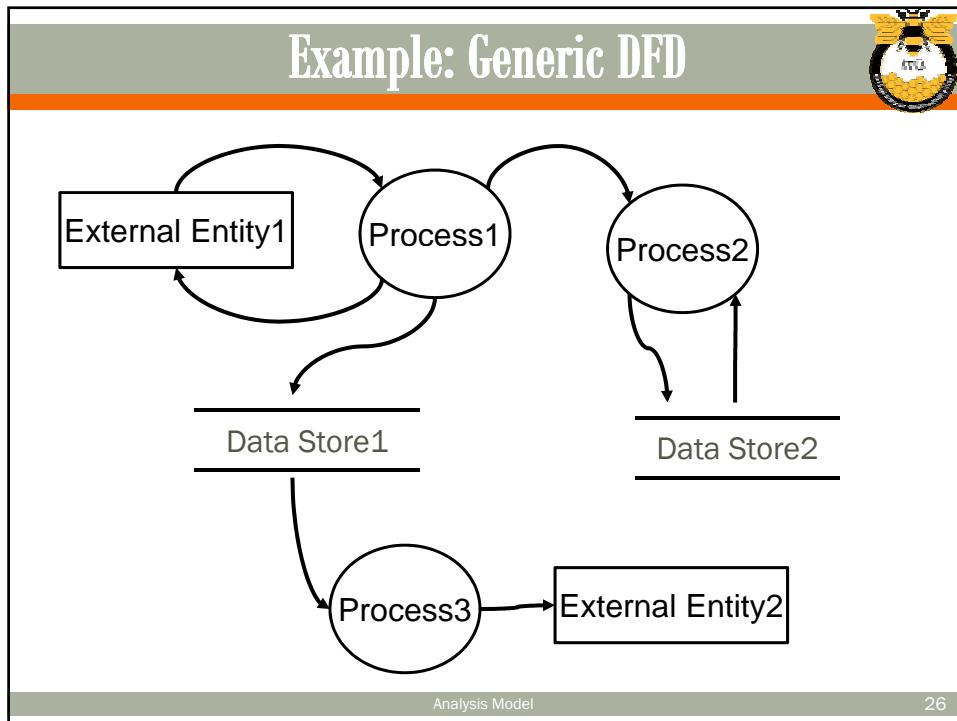
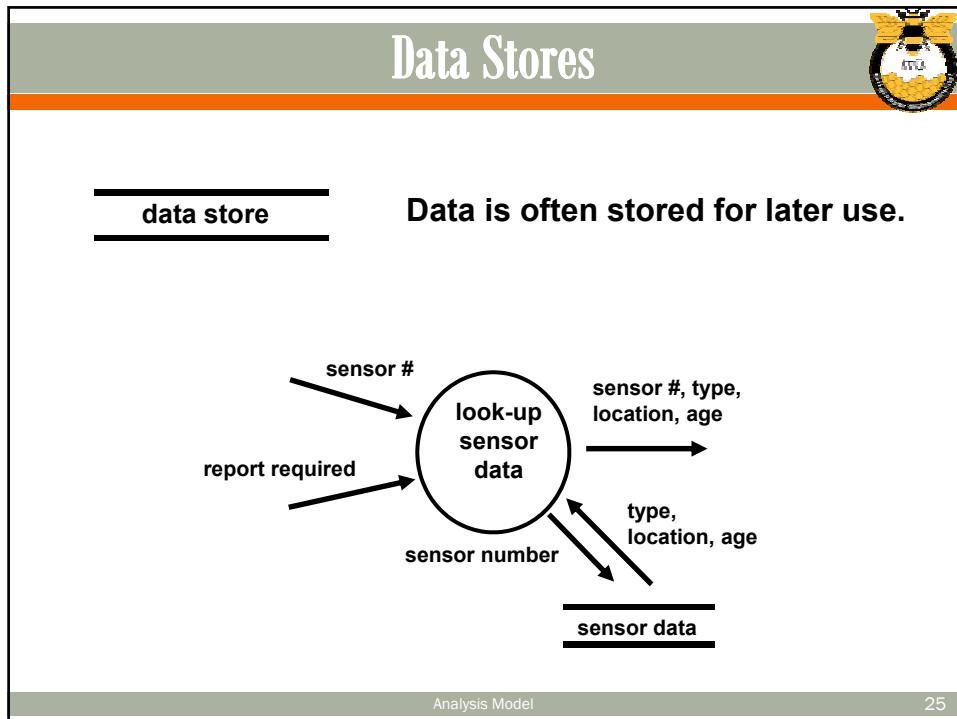


**Data flows through a system, beginning as input and being transformed into output.**

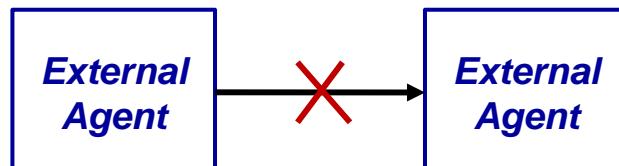


Analysis Model

24



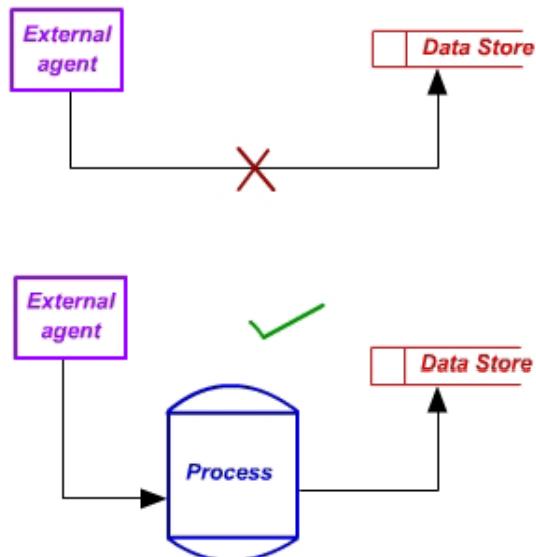
## DFD Rules (1)



Analysis Model

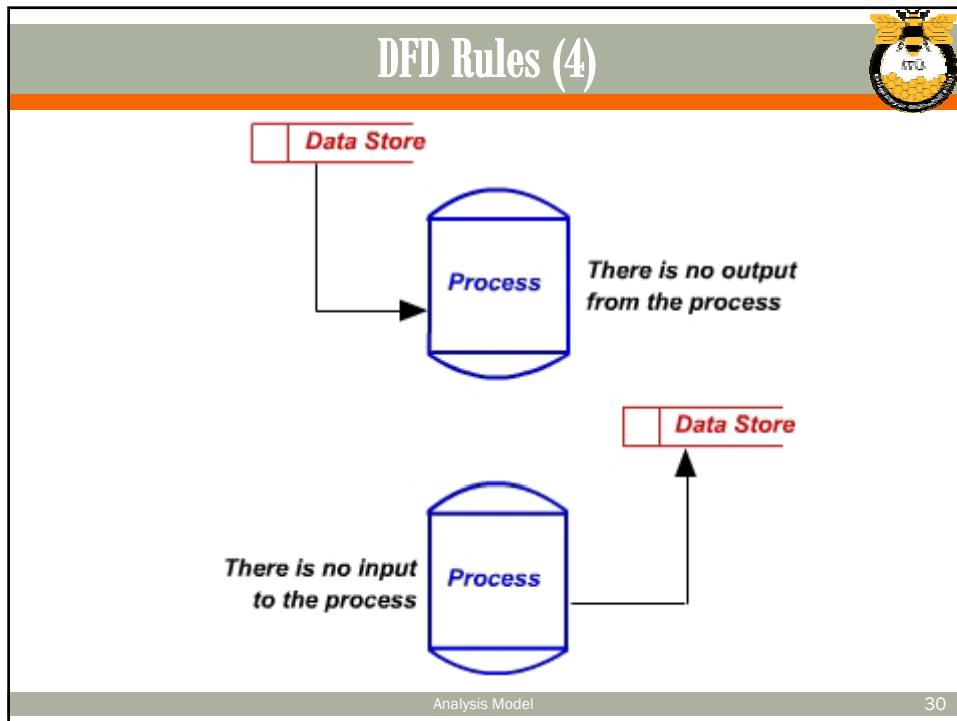
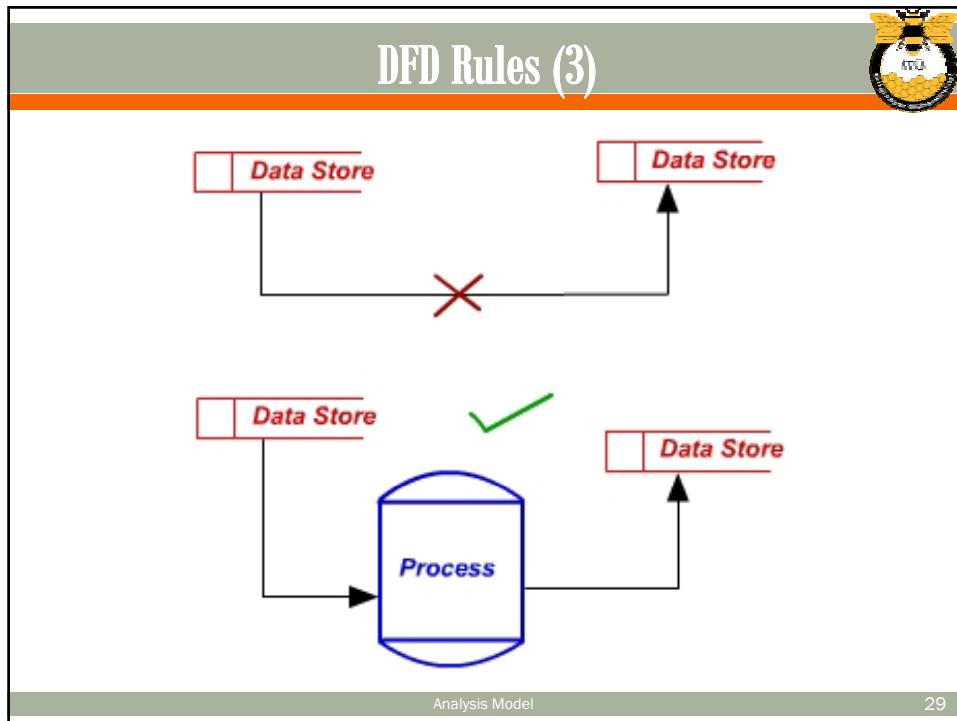
27

## DFD Rules (2)



Analysis Model

28



## 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

Analysis Model

31

### Example of Data Flow Refinement

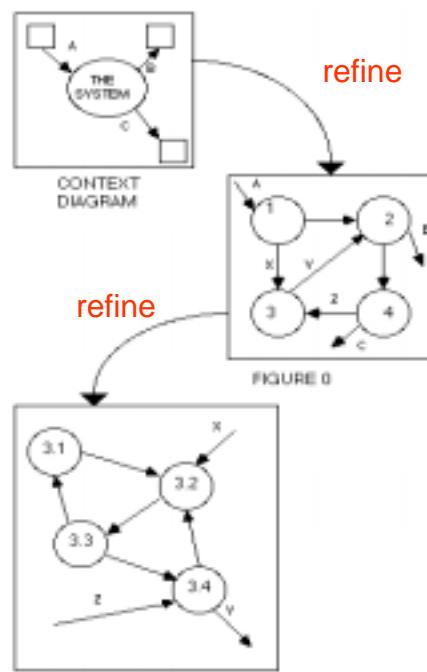
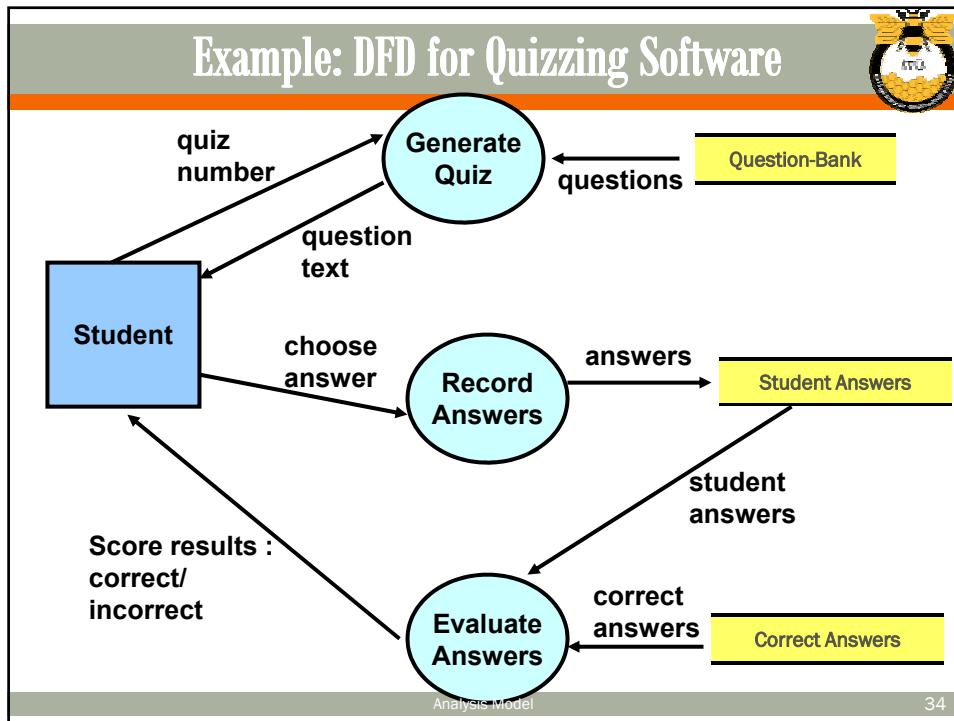
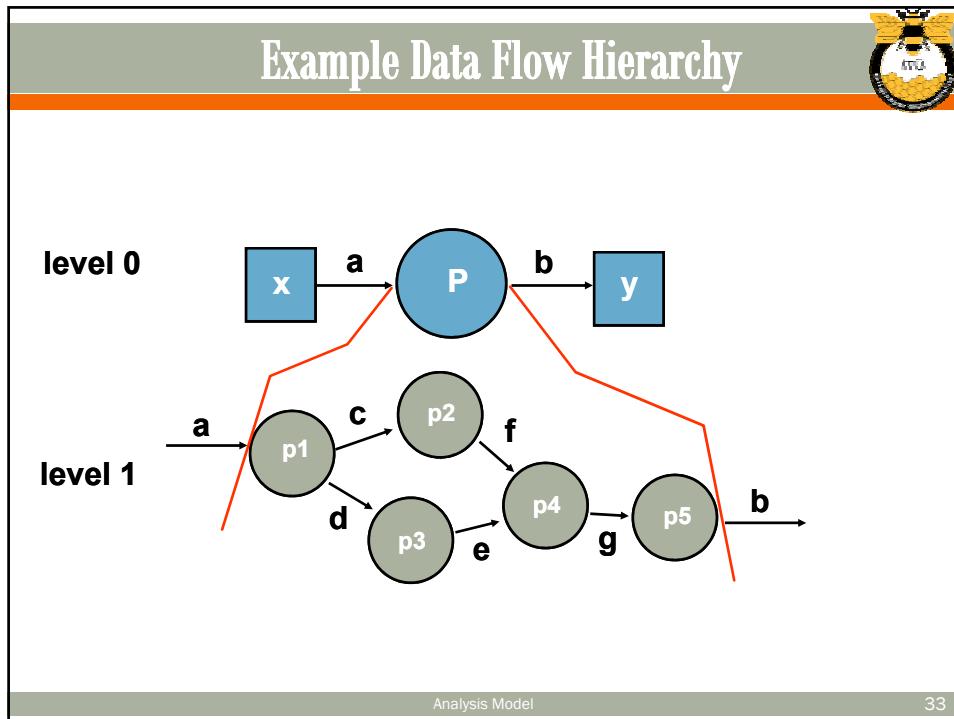
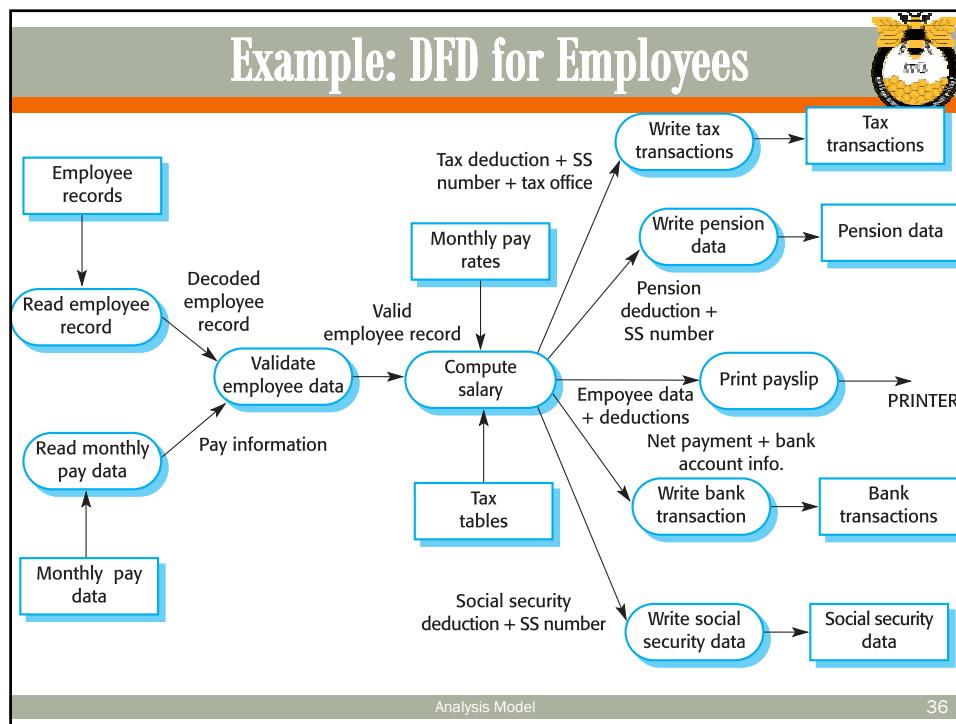
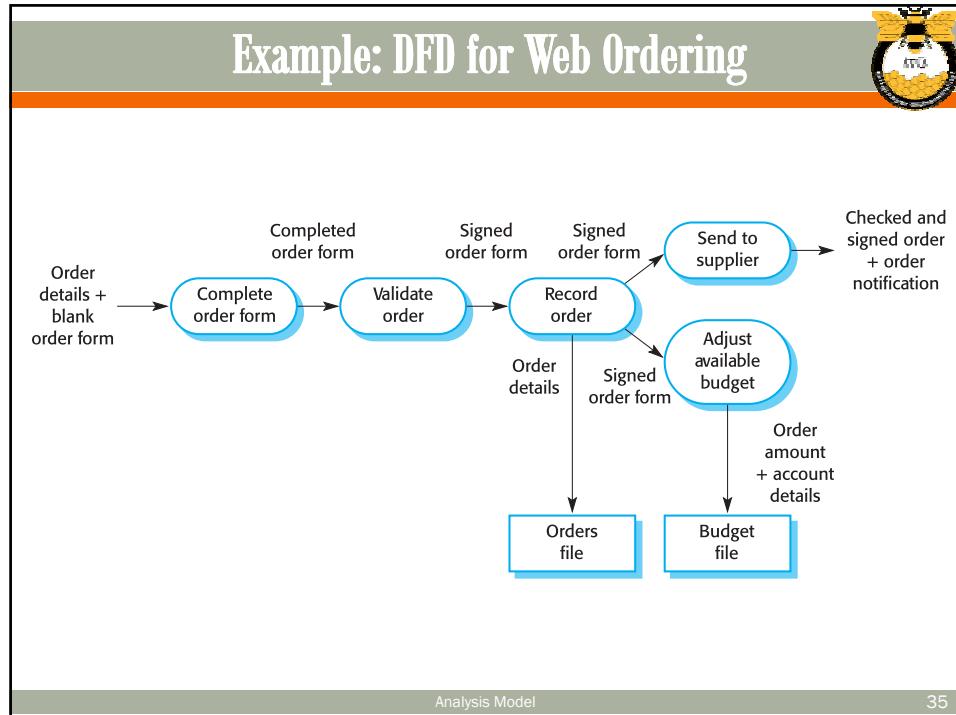
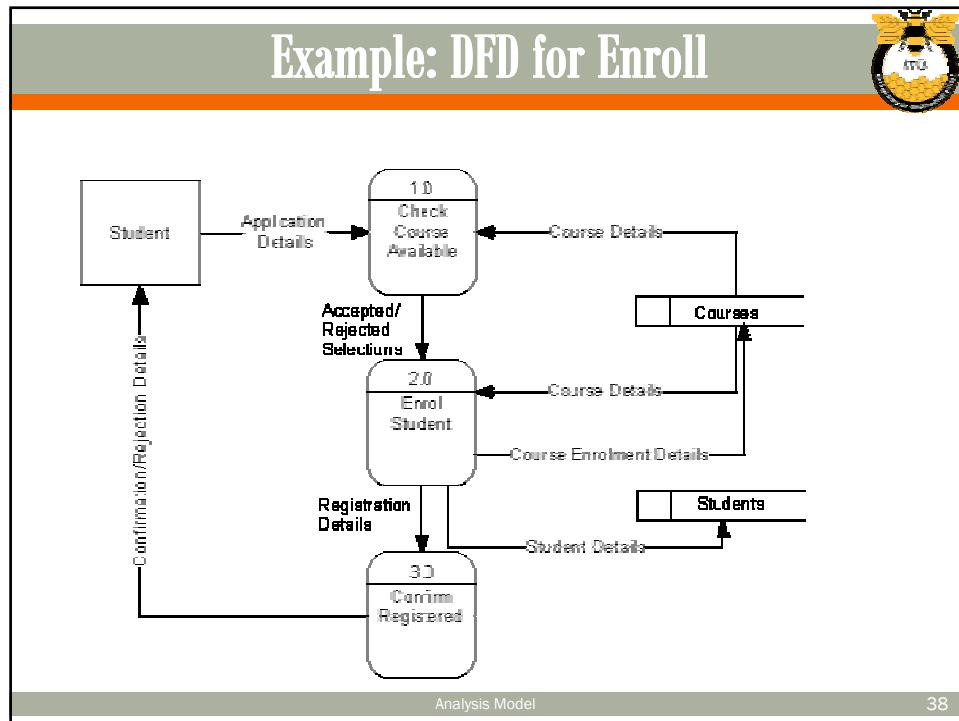
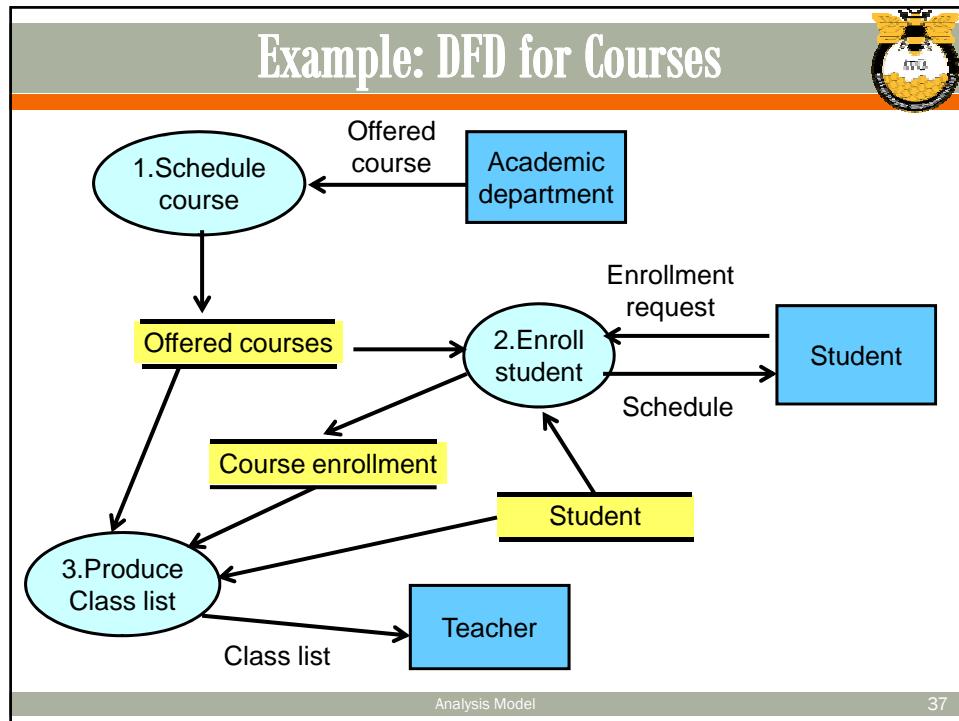


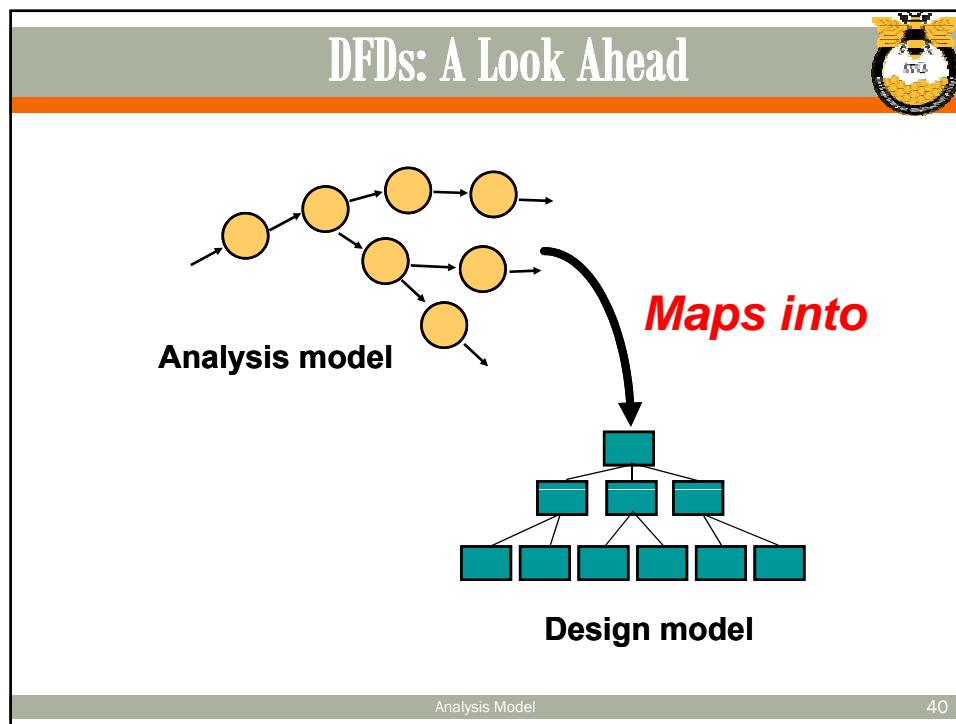
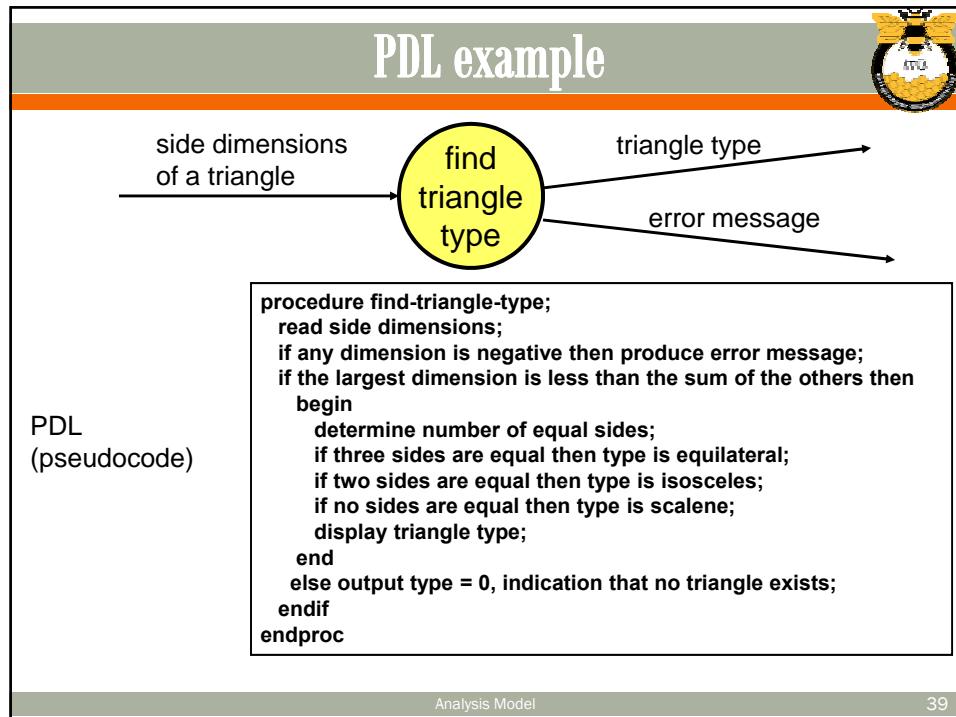
FIGURE 3

32









1. Data Model
2. Functional Model
3. Behavioural Model 

# The Behavioural Model

8.3

Analysis

## 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**.

Analysis Model

43

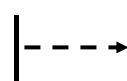
## Control Flow Diagrams



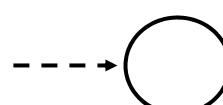
- » a dashed arrow entering a vertical bar is an input



- » 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



Analysis Model

44

## State Transition Diagrams

```

graph TD
    S1[STATE - 1] -- "CONDITION 0  
ACTION 0" --> S2[STATE - 2]
    S2 -- "CONDITION 1  
ACTION 1" --> S2
    S2 -- "CONDITION 2  
ACTION 2" --> S3[STATE - 3]
    S2 -- "CONDITION 4  
ACTION 4" --> S4[STATE - 4]
    S3 -- "CONDITION 3  
ACTION 3" --> S2
  
```

Analysis Model      45

**Example:**

Vending Machines  
Management Software

Analysis Model      46

## 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 (satış) 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.

## System Outline



### Location 1

- machine 1
- machine 2



**Service person 1**

### Location 2

- machine 1
- machine 2
- machine 3



**Service person 2**

### Warehouse

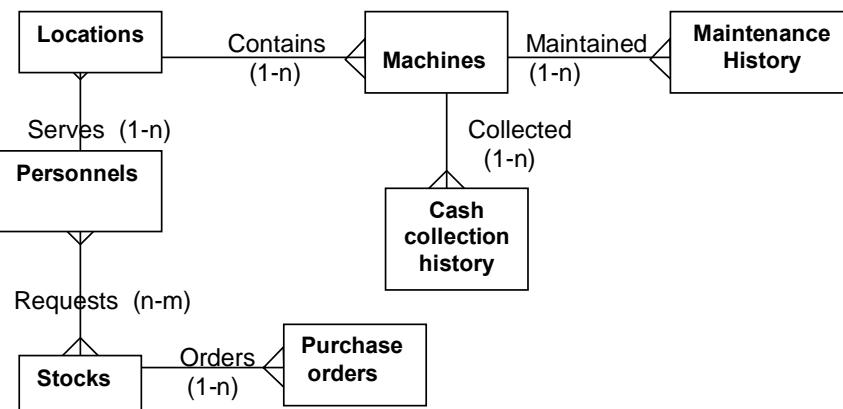
- food 1
- food 2
- food 3
- ...

## 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)



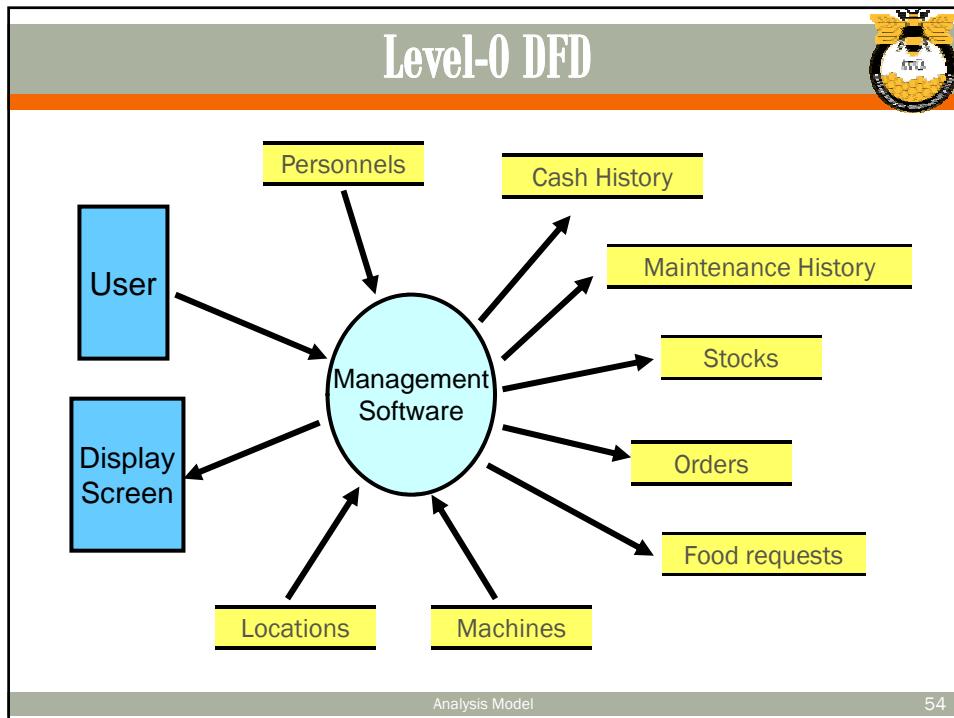
**Entities**

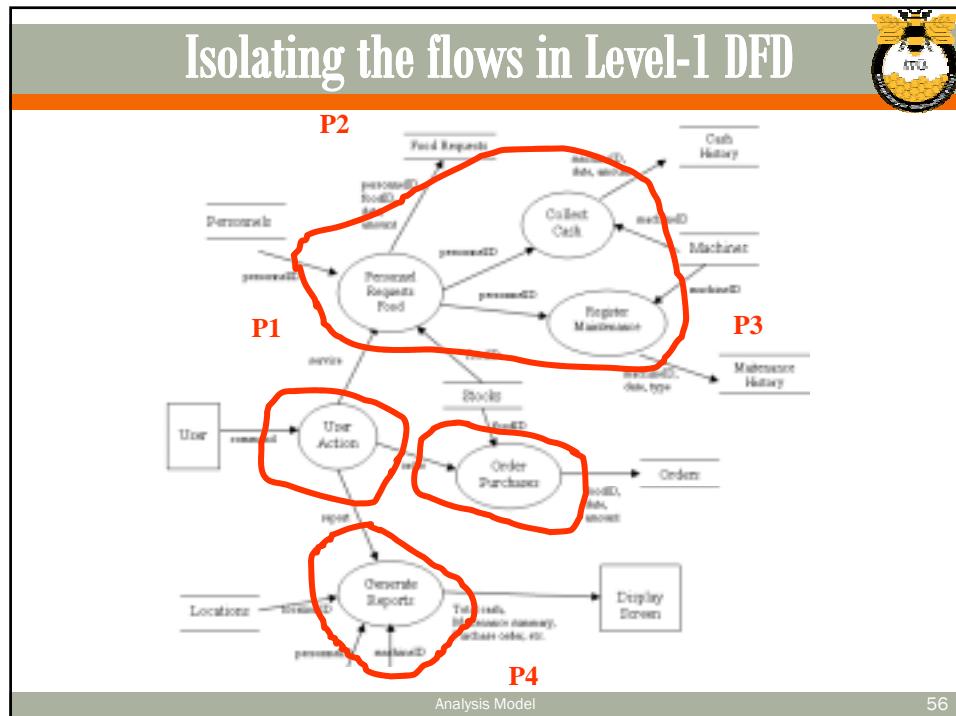
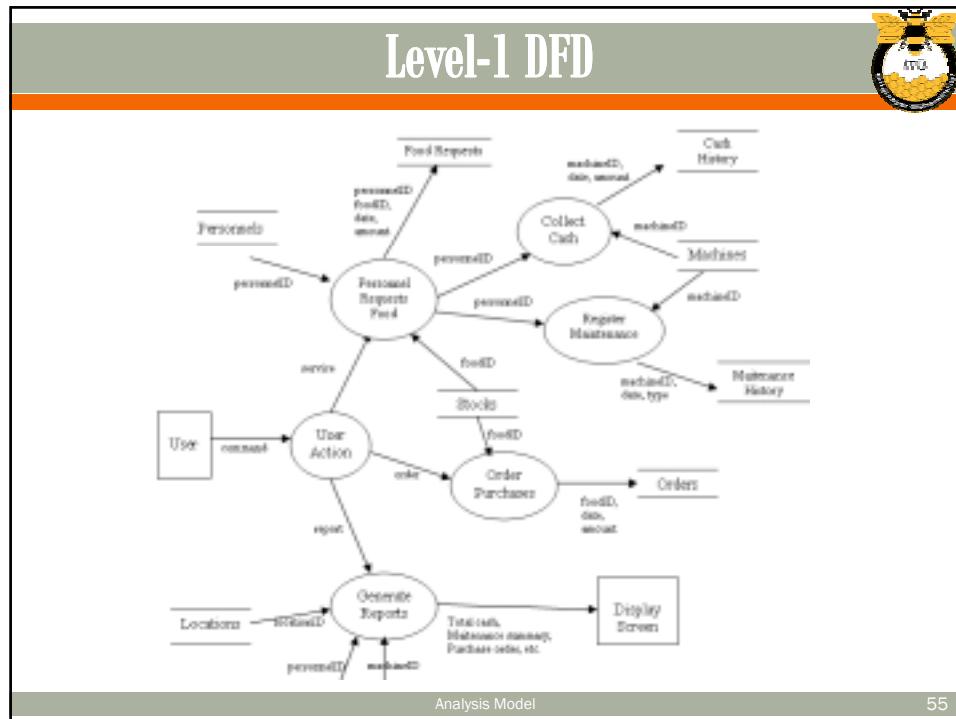


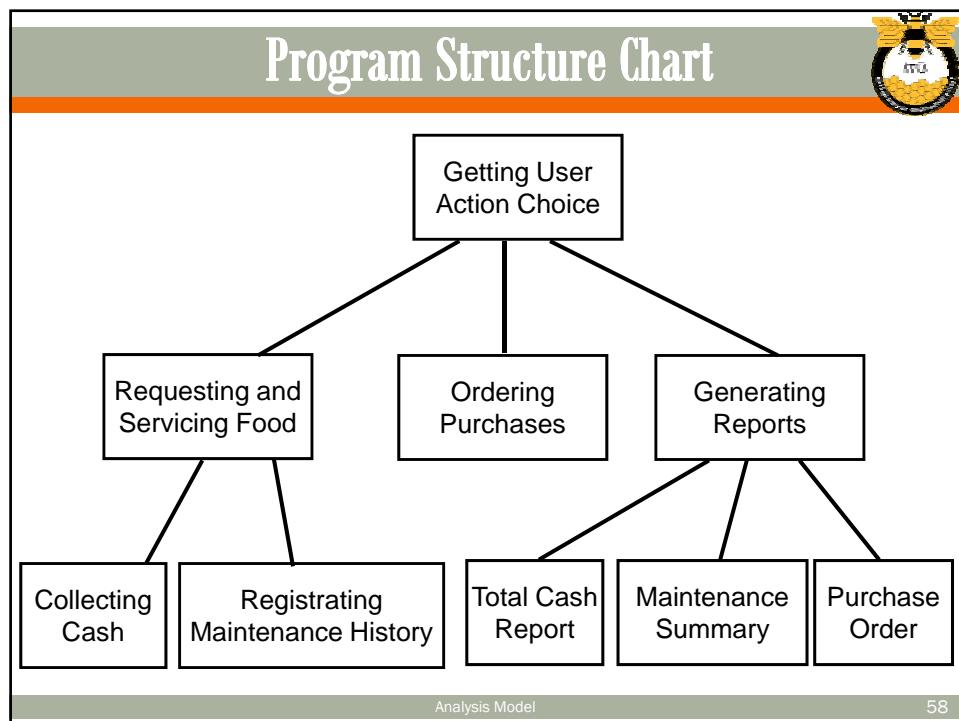
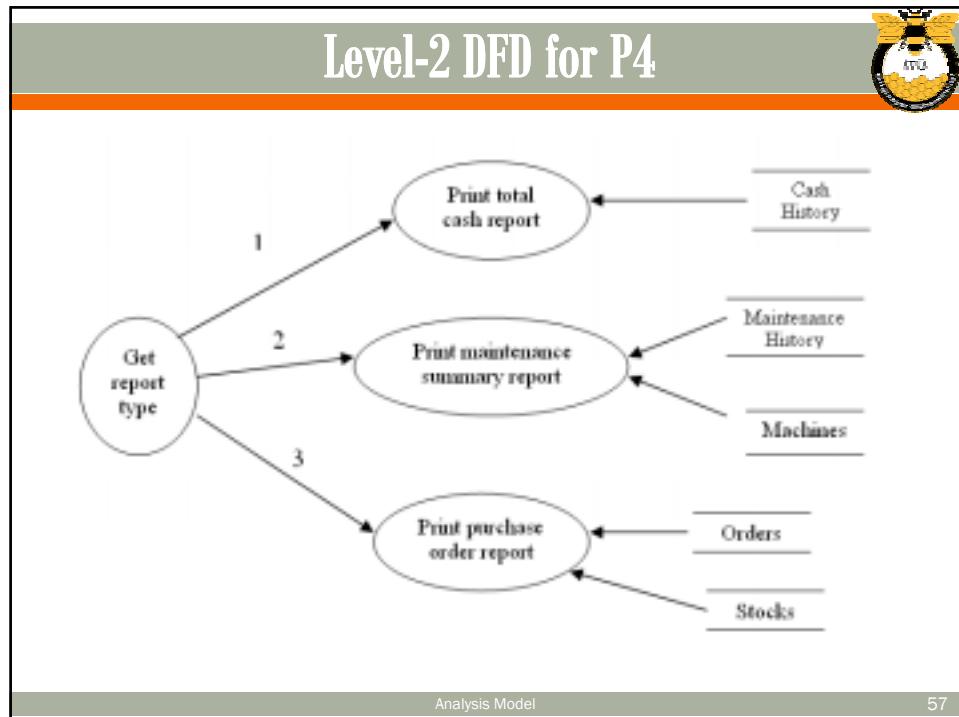
ENTITIY	DATA ITEMS
Locations	Location_ID, Address, Number of consumers, ServicePersonnel_ID
Machines	Machine_ID, Location_ID, Frequency of refilling
Personnels	Personnel_ID, Personnel name
Stocks	Food_ID, Food name, Current amount
Food_Requests	Personnel_ID, Food_ID, Date of request, Amount of request,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

Analysis Model

53









## Example:

### Technical Service Management Software

Analysis Model

59

## 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.
- » The followings are functional requirements:
- » 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.

Analysis Model

60

## 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)



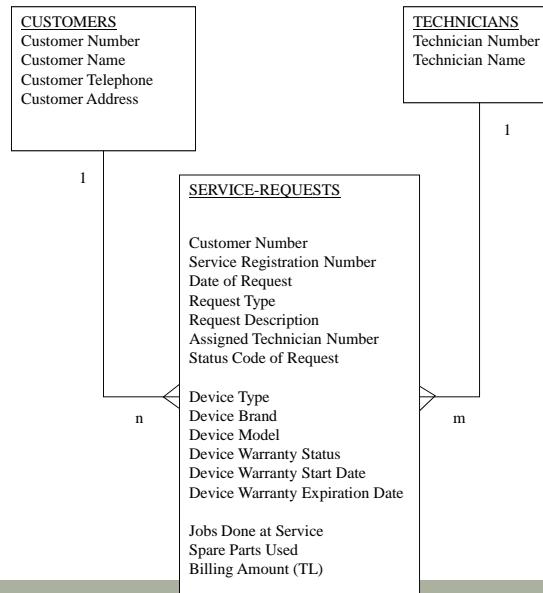
“Service Request Lists” should be available with different criteria:

- by service registration number
- by customer name
- by status code
- by device type
- by request type
- by date of request
- by technician name

Analysis Model

63

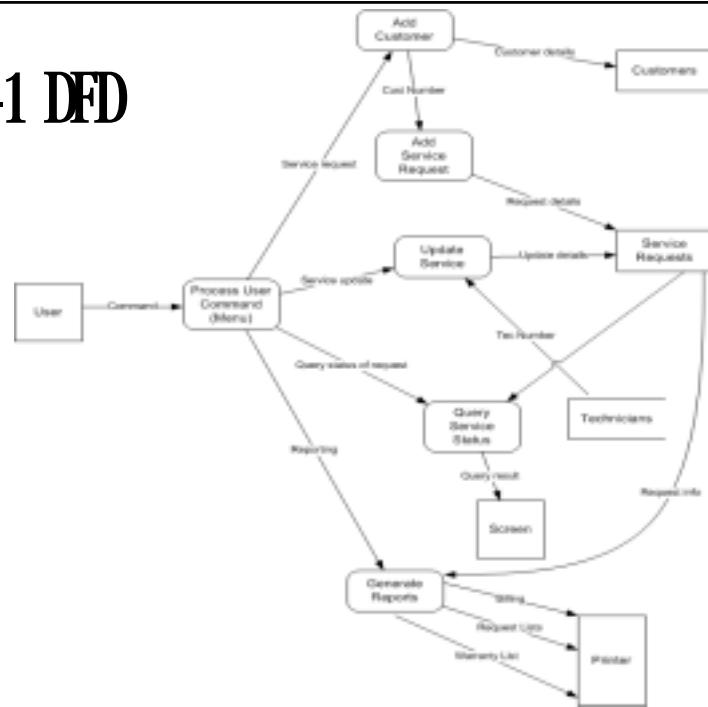
## Entity Relationship Diagram (ERD)



Analysis Model

64

## Level-1 DFD



## Program Structure Chart

