# Requirements Engineering

## Semi-Formal Specification:

### Structural Functional Requirements – Structured Analysis

- ❑Data Flow Diagrams

## Types of Requirements along different views

- Functional Requirements (FRs)
  - Structural Functional Requirements
    - Functional, i.e., Function-oriented
    - Informational. i.e., Information-oriented
  - Behavioral Functional Requirements
- Non-Functional Requirements (NFRs)

## Function Oriented Problem Analysis

- Creates a hierarchy of functions
  - Also called (process, activity, work-step, transactions, transforms, bubbles)
- Root is most abstract
- Leaf nodes the most detailed
- Most methods use data flow diagrams and dictionaries
- Examples
  - SRD structured requirements definition
  - SADT
  - Information Engineering
  - Modern structured Analysis
  - Problem statement Language

## Process Modeling

A data flow diagram (DFD) is a tool (and type of process model) that depicts the flow of data through a system and the work or processing performed by that system.

DFDs have become a popular tool for business process redesign.

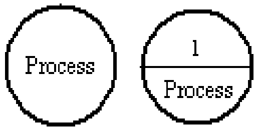
## Data Flow Diagrams

- Existed long before computers
- Show the flow of data through a system
- System
  - Organization
  - Company
  - A computer hardware system
  - A software system
- Icons
  - Data on the move – named arrows
  - Transformations of data – named bubbles
  - Sources and destinations of data – named rectangles (terminators)
  - Data in static storage – two parallel lines

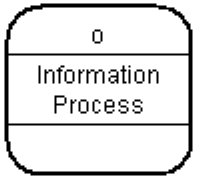
# Data Flow Diagram Notations

**Process**
A process transforms incoming data flow into outgoing data flow.

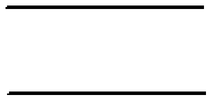
*Yourdon and Coad Process Notations*



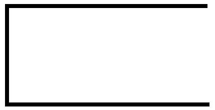
*Gane and Sarson Process Notation*



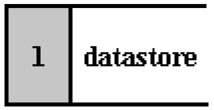
**Data Store**
Data stores are repositories of data in the system. They are sometimes also referred to as files.



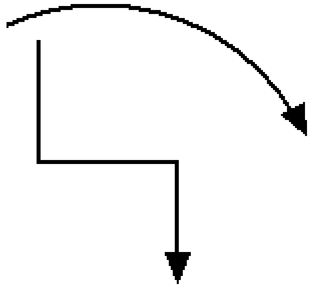
*Yourdon and Coad Datastore Notation*



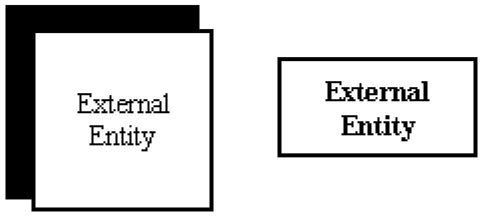
*Gane and Sarson Datastore Notations*



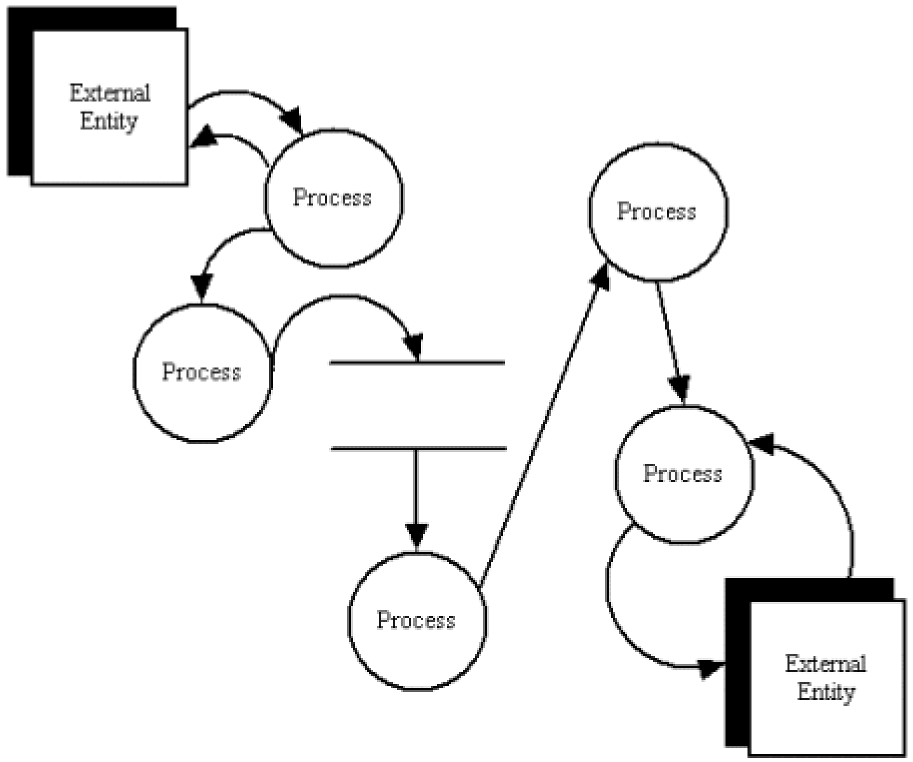
**Dataflow**
Dataflows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.



**External Entity**
External entities are objects outside the system, with which the system communicates. External entities are sources and destinations of the system's inputs and outputs.

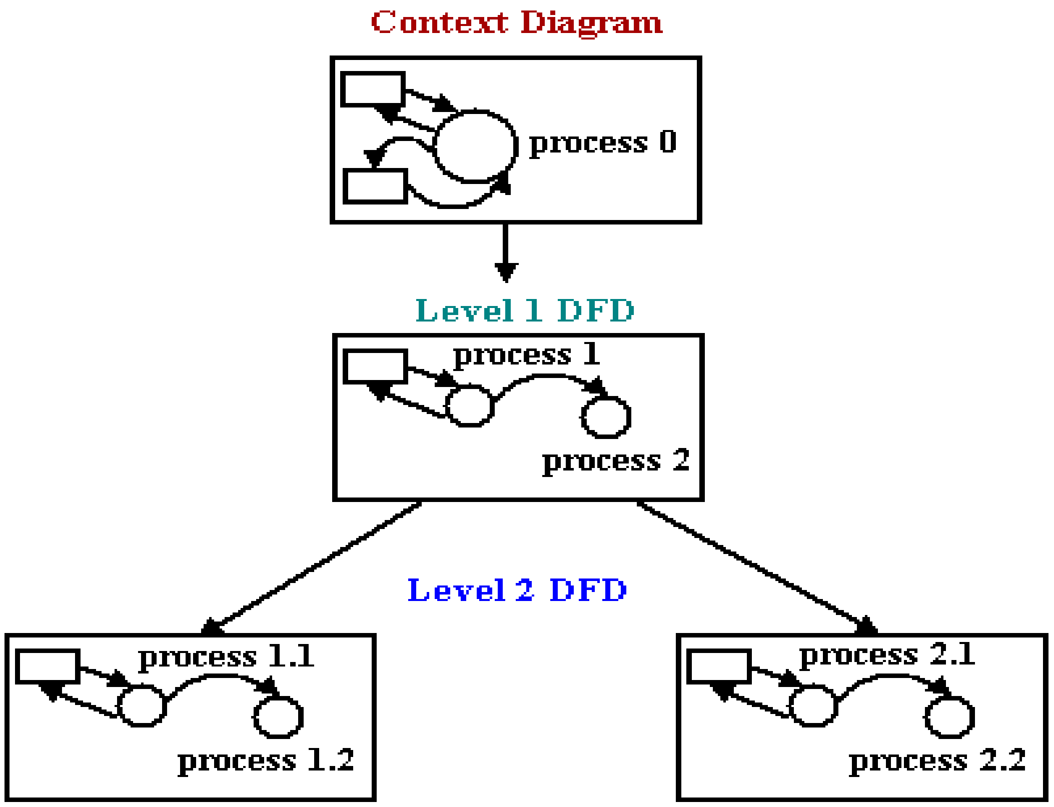


# DataFlow Diagrams



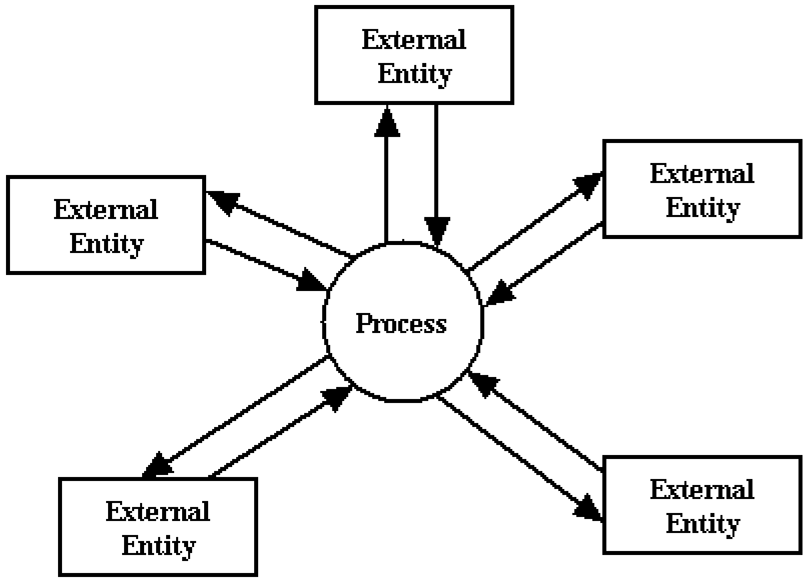
# Data Flow Diagram Layers

- Data flow diagrams are drawn in several nested layers
- A single process node on a high level diagram can be expanded to show a more detailed data flow diagram.
- Draw the context diagram first, followed by various layers of data flow diagrams.



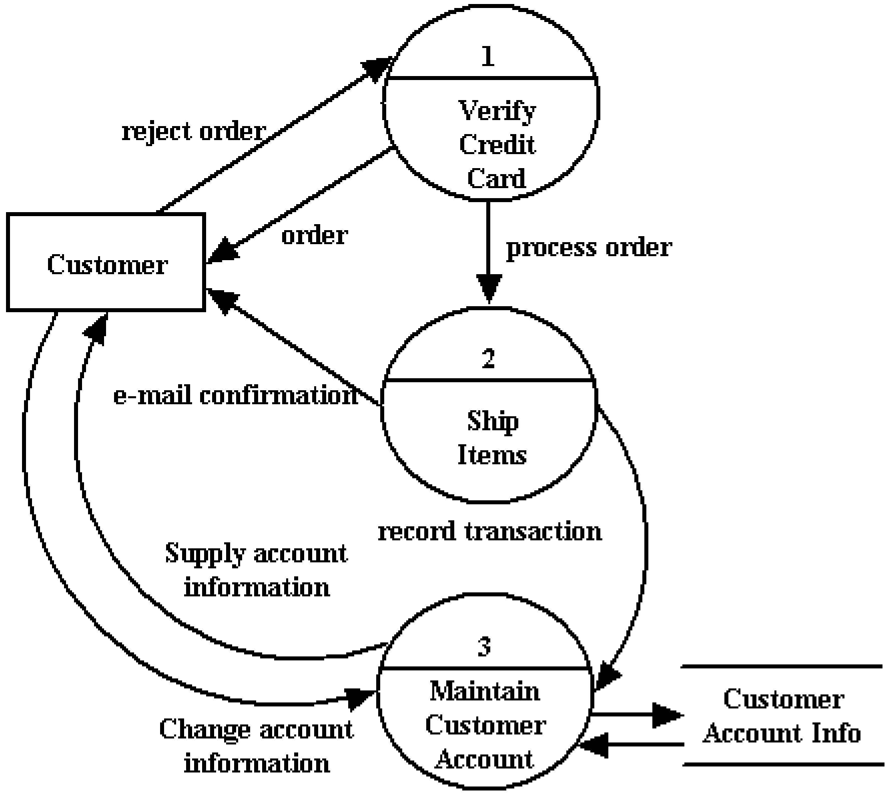
## Context Diagrams

- A context diagram is a top level (also known as Level 0) data flow diagram. It only contains one process node (process 0) that generalizes the function of the entire system in relationship to external entities.

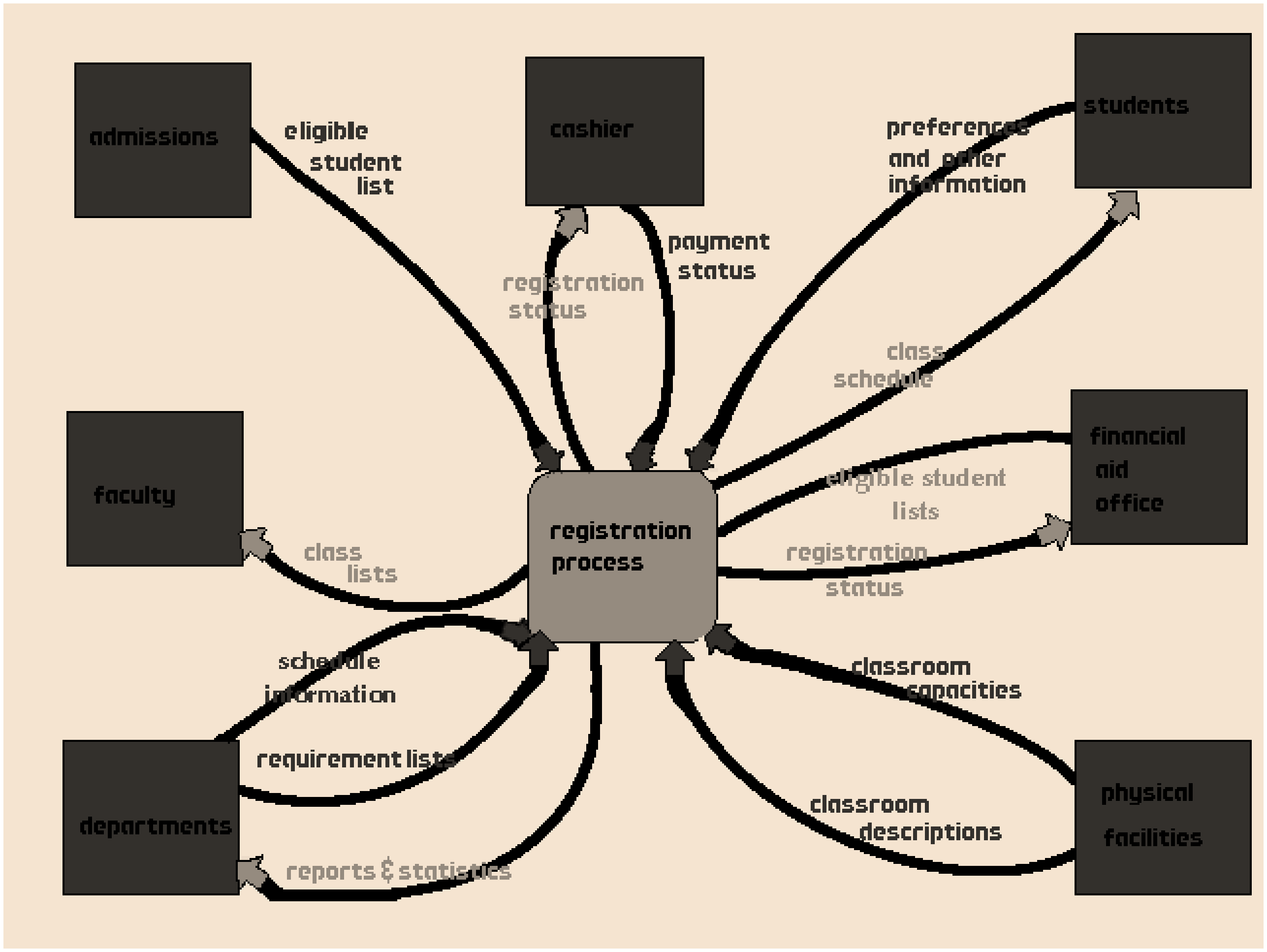


# DFD levels

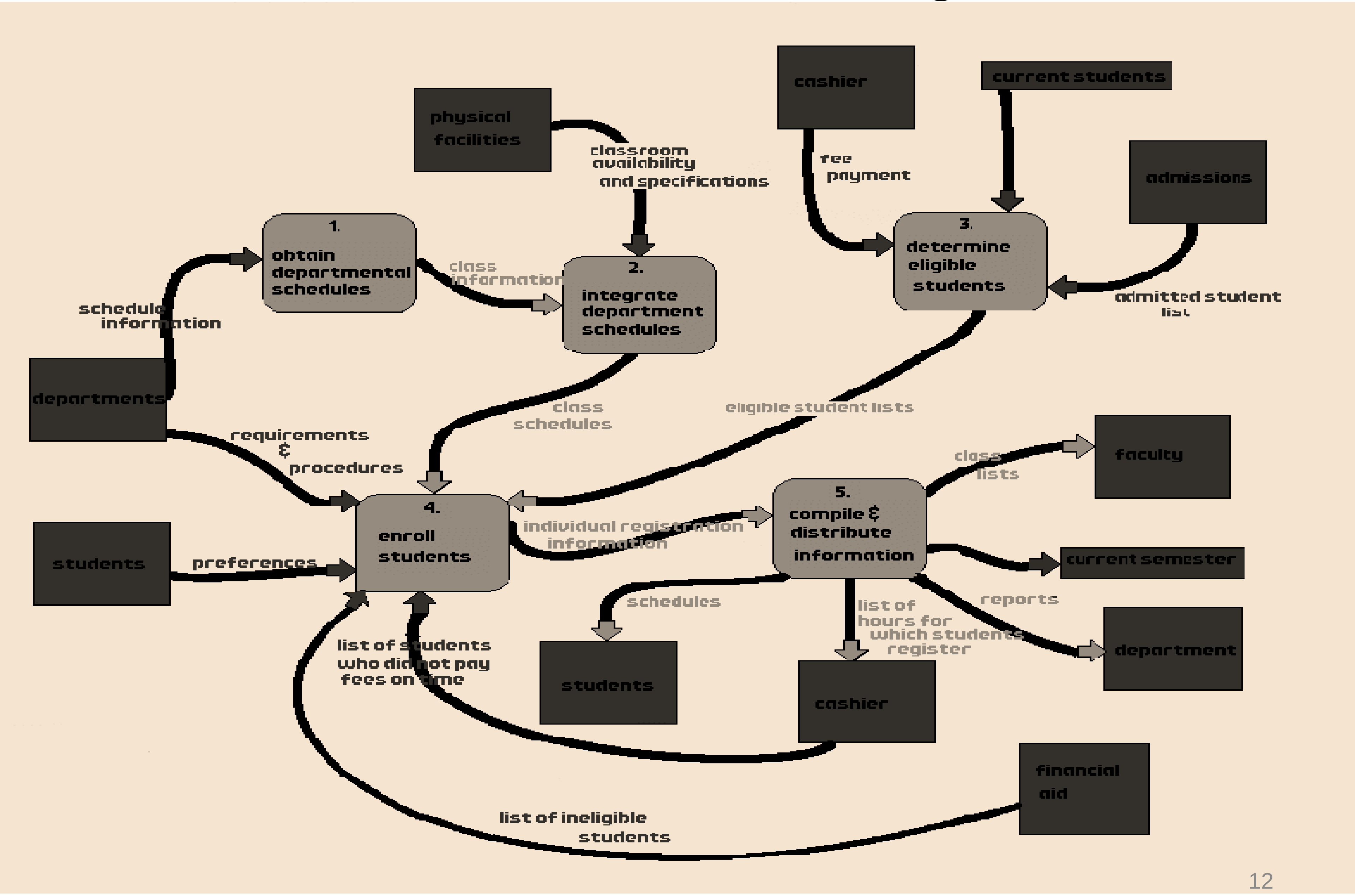
The first level DFD shows the main processes within the system. Each of these processes can be broken into further processes until you reach pseudocode.



## Context Diagram- Registration

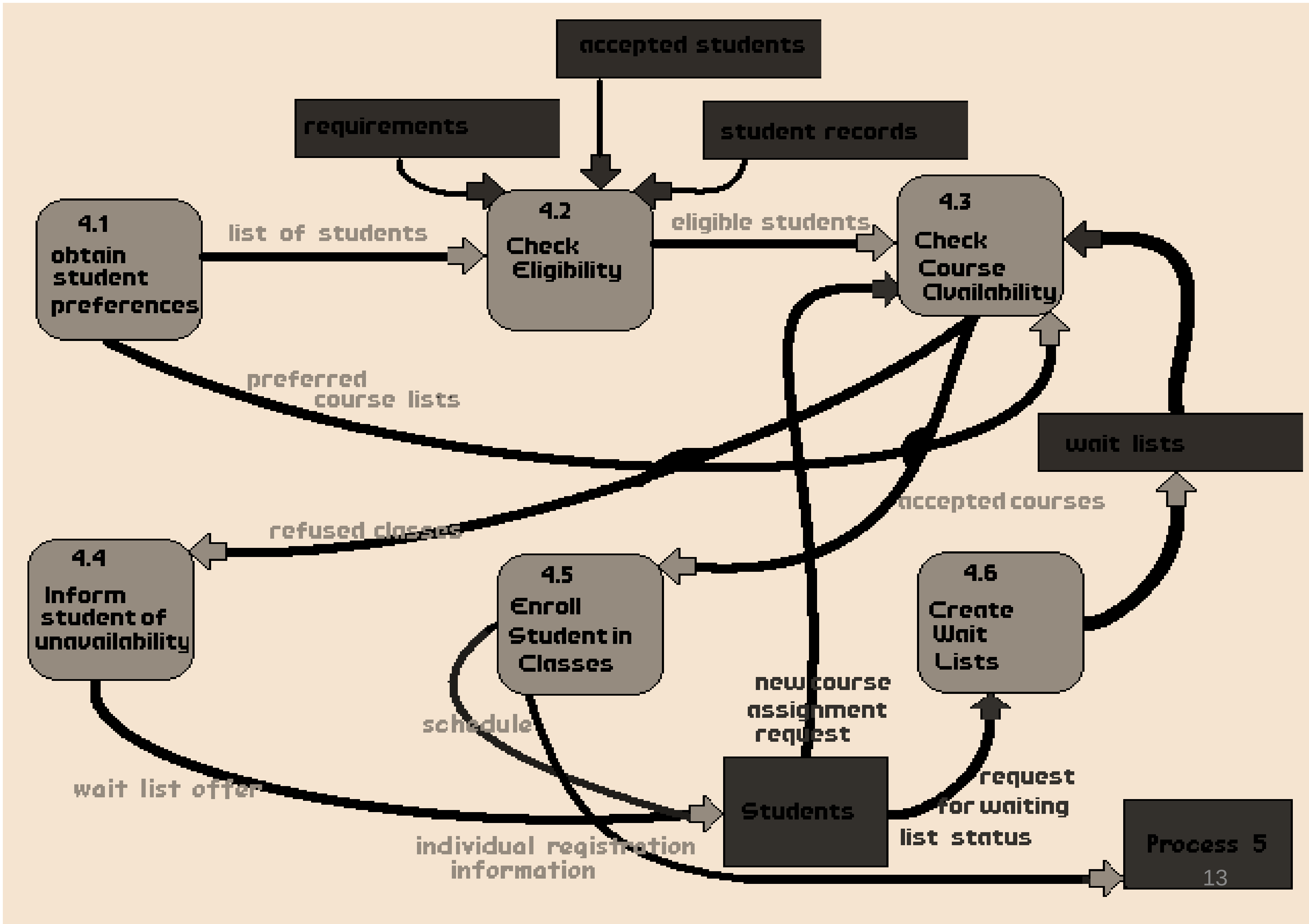


# Level 0 Data Flow Diagram



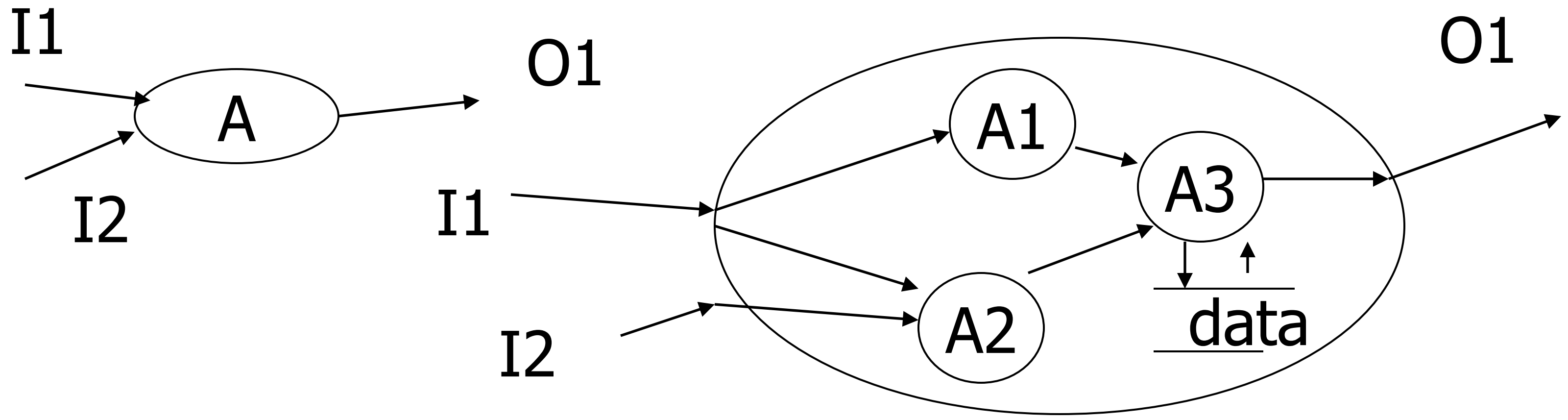
physical facilities
classroom availability and specifications
cashier
fee payment
current students
admissions
admitted student list
1. obtain departmental schedules
class information
2. integrate department schedules
3. determine eligible students
schedule information
departments
class schedules
eligible student lists
requirements & procedures
faculty
class lists
4. enroll students
5. compile & distribute information
individual registration information
students
preferences
current semester
schedules
list of hours for which students register
reports
list of students who did not pay fees on time
students
cashier
department
financial aid
list of ineligible students

# Explosion of Process 4



## Data Flow Diagrams

- Rules
  - All names must be unique
  - Not a flow chart – no order implied
  - No logical decisions
  - Don't get bogged down in detail
- Leveling
  - Preserve the number of inputs and outputs between the levels



## Difference between DFD and Flowcharts

- Processes on DFDs can operate in parallel (at-the-same-time)
  - Processes on flowcharts execute one at a time

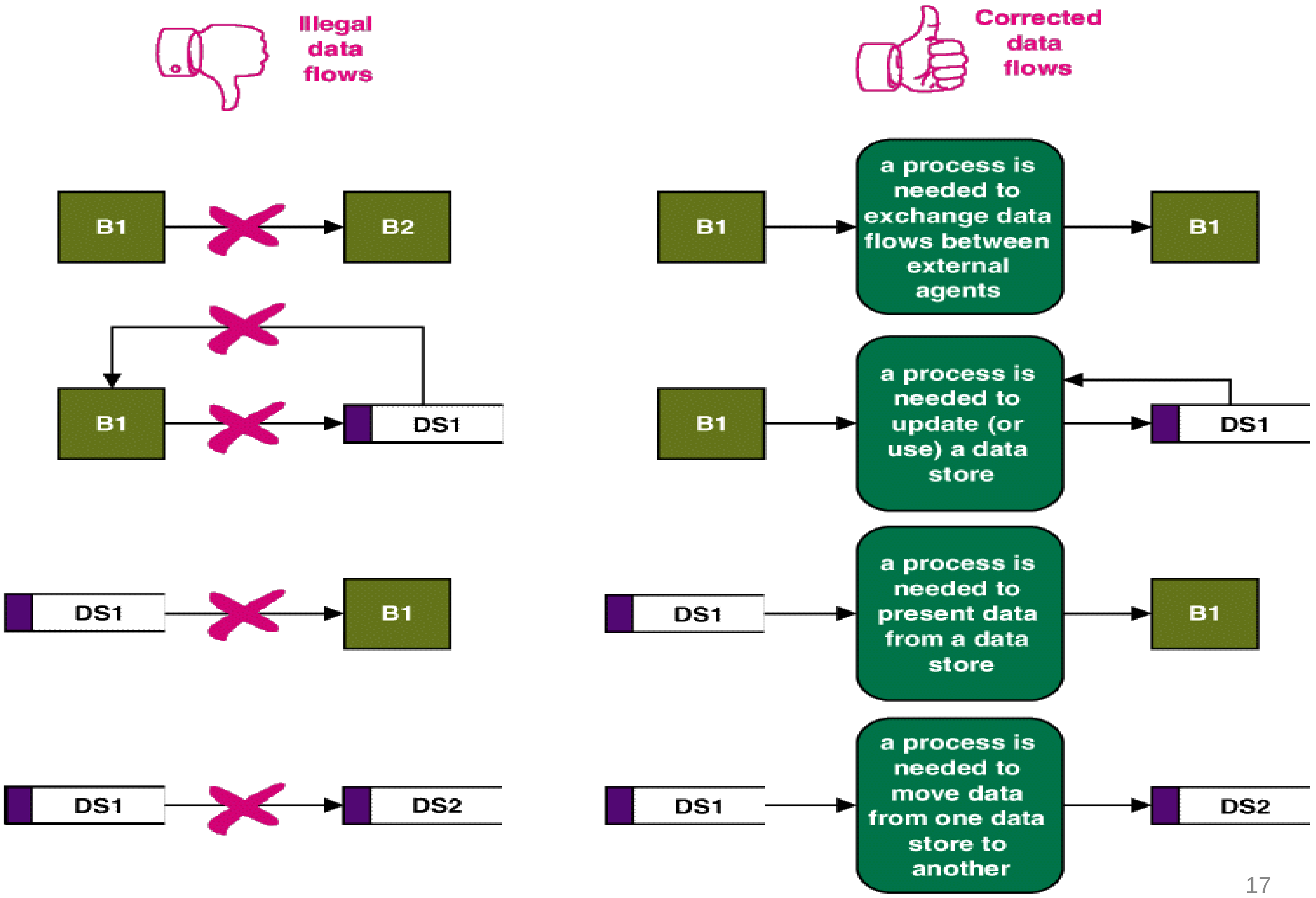
- DFDs show the flow of data through a system
  - Flowcharts show the flow of control (sequence and transfer of control)

- Processes on one DFD can have dramatically different timing
  - Processes on flowcharts are part of a single program with consistent timing

## Homework -

- Draw a DFD for an ATM

# Illegal Data Flows


Illegal data flows
Corrected data flows
B1
B2
B1
DS1
DS1
B1
DS1
DS2
B1
a process is needed to exchange data flows between external agents
B1
B1
a process is needed to update (or use) a data store
DS1
DS1
a process is needed to present data from a data store
B1
DS1
a process is needed to move data from one data store to another
DS2

# Structured English context

| Construct | Sample Template |
|---|---|
| **Sequence of steps –** unconditionally perform a sequence of steps. | [ Step 1 ] [ Step 2 ] … [ Step n ] |
| **Simple condition steps** – if the specified condition is true, then perform the first set of steps. Otherwise, perform the second set of steps. Use this construct if the condition has only two possible values. (Note: The second set of conditions is optional.) | **If** [ truth condition ] **then** [ sequence of steps or other conditional steps ] **else** [ sequence of steps or other conditional steps ] ~~**End If**~~ |
| **Complex condition steps** – test the value of the condition and perform the appropriate set of steps. Use this construct if the condition has more than two values. | **Do the following based on** [ condition ]: **Case 1: If** [ condition] = [value] then [ sequence of steps or other conditional steps ] **Case 2: If** [ condition] = [value] then [ sequence of steps or other conditional steps ] … **Case n: If** [ condition] = [value] then [ sequence of steps or other conditional steps ] ~~**End Case**~~ |

## Structured English Rules

Condition stubs

**Multiple conditions –** test the value of multiple conditions to determine the correct set of steps.

Use a **decision table** instead of nested if-then-else Structured English constructs to simplify the presentation of complex logic that involves combinations of conditions.

*A* **decision table** is a *tabular presentation of complex logic in which rows represent conditions and possible steps, and columns indicate which combinations of conditions result in specific steps.*

| DECISION TABLE | Rule | Rule | Rule | Rule |
|---|---|---|---|---|
| [ Condition ] | value | value | value | value |
| [ Condition ] | value | value | value | value |
| [ Condition ] | value | value | value | value |
| [ Sequence of actions or conditional actions ] | X | | | |
| [ Sequence of actions or conditional actions ] | | X | X | |
| [ Sequence of actions or conditional actions ] | | | | X |

*Although it isn't a Structured English construct, a decision table can be named, and referenced within a Structured English procedure.*

Action Stubs

## Decision Table Example

| ***Condition Stub*** | | ***Condition Entry*** | | | | | |
|---|---|---|---|---|---|---|---|
| **If** | Customer is bookstore | Y | Y | N | N | N | N |
| | Order size > 6 copies | Y | N | N | N | N | N |
| | Customer is librarian/individual | | | Y | Y | Y | Y |
| | Order size 50 copies or more | | | Y | N | N | N |
| | Order size 20 to 49 copies | | | | Y | N | N |
| | Order size 6 – 19 copies | | | | | Y | N |
| | | | | | | | |
| Then | Allow a 25% discount | X | | | | | |
| | Allow a 15% discount | | | X | | | |
| | Allow 10% discount | | | | X | | |
| | Allow a 5 % discount | | | | | X | |
| | Allow 0% discount | | X | | | | X |
| | ***Action Stub*** | ***Action Entries*** | | | | | |

## Structured English constructs

| | |
|---|---|
| **One-to-many iteration –** Repeat the set of steps until the condition is false.<br><br>Use this construct if the set of steps must be performed at least once, regardless of the condition's initial value. | **Repeat the following until** [truth condition]:<br>[ sequence of steps or conditional steps ]<br>~~**End Repeat**~~ |
| **Zero-to-many iteration –** Repeat the set of steps until the condition is false.<br><br>Use this construct if the set of steps are conditional based on the condition's initial value. | **Do While** [truth condition]:<br>[ sequence of steps or conditional steps ]<br>~~**End Do**~~<br><br>**- OR -**<br><br>**For** [truth condition]:<br>[ sequence of steps or conditional steps ]<br>~~**End For**~~ |

# Structured English

Structured English is a language and syntax, based on the relative strengths of structured programming and natural English, for specifying the underlying logic of elementary processes on DFDs.

```
1. For each CUSTOMER NUMBER in the data store CUSTOMERS:
   a. For each LOAN in the data store LOANS that matches the above
      CUSTOMER NUMBER:
      1) Keep a running total of NUMBER OF LOANS for the
         CUSTOMER NUMBER.
      2) Keep a running total of the ORIGINAL LOAN PRINCIPAL for the
         CUSTOMER NUMBER.
      3) Keep a running total of CURRENT LOAN BALANCE for the
         CUSTOMER NUMBER.
      4) Keep a running total of AMOUNTS PAST DUE for the
         CUSTOMER NUMBER.
   b. If the TOTAL AMOUNTS PAST DUE for the CUSTOMER NUMBER
         is greater than $100.00 then:
         1) Write the CUSTOMER NUMBER and all their data attributes
            as described in the data flow LOANS AT RISK.
      Else
         1) Exclude the CUSTOMER NUMBER and data from the data
            flow LOANS AT RISK.
```

## Data Dictionaries

- Used to augment the Data Flow Diagrams
- Repository
- Layout
  - Name of the item
  - Alias
  - Description/Purpose
  - Related data items
  - Range of values
  - Data flows
  - Data structure definition/form

# Data Structures

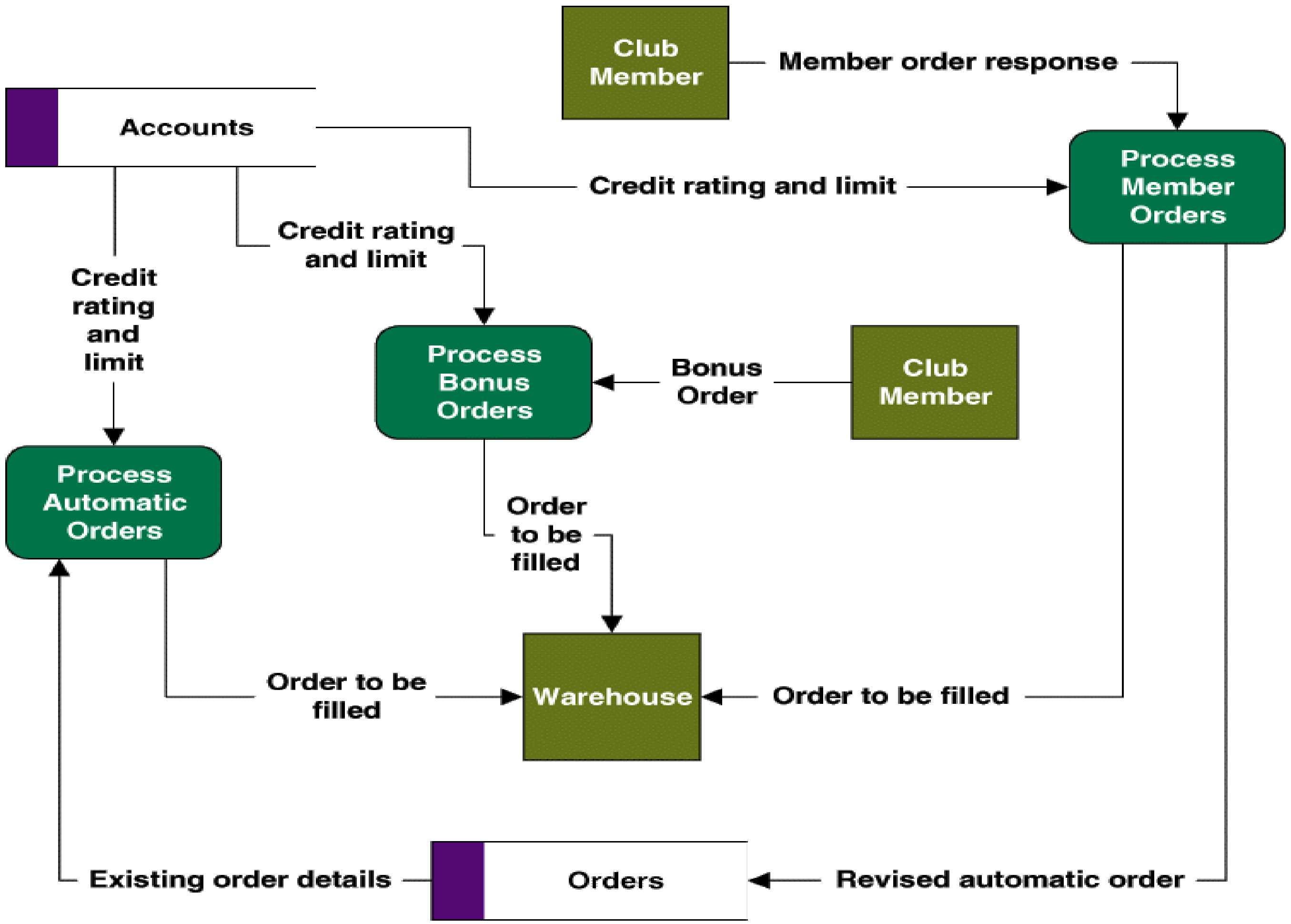
- Are specific arrangements of data attributes that define a single instance of a data flow
  - A sequence that occur one after another
  - A selection of one or more attributes from a set
  - A repetition of one or more attributes
- Most common data structure notation is Boolean algebraic notation

| | |
|---|---|
| = | "Consists of" or "is composed of" |
| + | Means and and designates sequence |
| [. . .] | Only one of the attributes may be present – ***selection*** - Attributes separated by commas<br>Either/or |
| {. . .} | Attributes may occur many times – ***repetition*** - Attributes separated by commas |
| (. . .) | Attributes in side are optional no value for some of the data flows |

## Example Data Structure

| DATA STRUCTURE | ENGLISH ENTERPRETATION |
|---|---|
| ORDER=<br>ORDER NUMBER +<br>ORDER DATE+<br>[ PERSONAL CUSTOMER NUMBER,<br>CORPORATE ACCOUNT NUMBER]+<br>SHIPPING ADDRESS=ADDRESS+<br>(BILLING ADDRESS=ADDRESS)+<br>1 {PRODUCT NUMBER+<br>PRODUCT DESCRIPTION+<br>QUANTITY ORDERED+<br>PRODUCT PRICE+<br>PRODUCT PRICE SOURCE+<br>EXTENDED PRICE } N+<br>SUM OF EXTENDED PRICES+<br>PREPAID AMOUNT+<br>(CREDIT CARD NUMBER+EXPIRATION DATE)<br>(QUOTE NUMBER) | An instance of ORDER consists of:<br>ORDER NUMBER and<br>ORDER DATE and<br>Either PERSONAL CUSTOMER NUMBER<br>or CORPORATE ACCOUNT<br>NUMBER<br>and SHIPPING ADDRESS (which is equivalent<br>to ADDRESS)<br>and optionally: BILLING ADDRESS (which is<br>equivalent to ADDRESS)<br>and one or more instances of:<br>PRODUCT NUMBER and<br>PRODUCT DESCRIPTION<br>and<br>QUANTITY ORDERED and<br>PRODUCT PRICE and<br>PRODUCT PRICE SOURCE<br>and<br>EXTENDED PRICE<br>and SUM OF EXTENDED PRICES and<br>PREPAID AMOUNT and<br>optionally: both CREDIT CARD NUMBER and<br>EXPIRATION DATE |
| ADDRESS=<br>(POST OFFICE BOX NUMBER)+<br>STREET ADDRESS+<br>CITY+<br>[STATE, MUNICIPALITY]+<br>(COUNTRY)+<br>POSTAL CODE | An instance of ADDRESS consists of:<br>optionally: POST OFFICE BOX NUMBER and<br>STREET ADDRESS and<br>CITY and<br>Either STATE or MUNICIPALITY<br>and optionally: COUNTRY<br>and POSTAL CODE |

# A Simple Process Model


Club Member
Member order response
Accounts
Credit rating and limit
Process Member Orders
Credit rating and limit
Credit rating and limit
Process Bonus Orders
Bonus Order
Club Member
Process Automatic Orders
Order to be filled
Order to be filled
Warehouse
Order to be filled
Existing order details
Orders
Revised automatic order