

1.3.4.2.1 Reports from data dictionary

- Documentation warehouse
 - Relationships should also be well described
- Reports include
 - Alphabetized list of elements
 - particular data element and a list of all data flows and stores that use it
 - A detailed report of any component

1.3.4.3 Tools for process description

- Process desc tools
 - Show processing details for business logic that are not shown in DFD
 - decision tables
 - decision trees
 - Processes for functional primitives broken down even further
 - Process called Modular design.

1.3.4.3.1 Modular Design

Flowcharts

- Mostly use modular design to show logic
- 3 logical Control Structures
 - Use symbols

→ data flow



process



condition

- 3 Structures

- Sequence
- Selection
- Iteration
- Single entry & exit-point

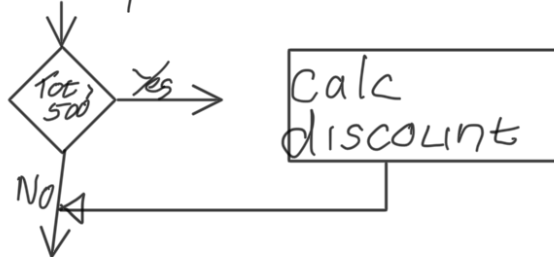
- Sequence

- 1 or more processes may be a sub-process containing other logical structures.



- Selection

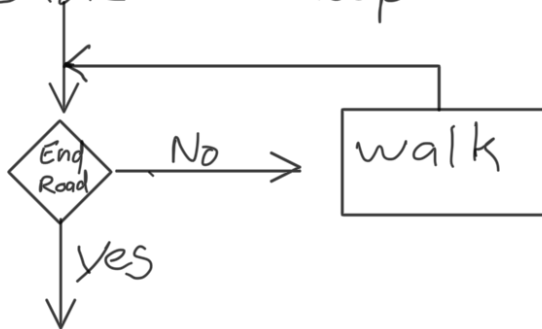
- Basic if statements



- depends on result

- Iteration

- Basic while loop



- repeat till change in condition

1.3.4.3.2 Structure English

- describe processing logic
- Use loops discussed earlier
- Use indentation
- Use limited vocab
 - describe processing rules
 - Or in data dictionary

For each ITEM

if PRICE is greater than 150

if PERSON is MEMBER

DISCOUNT is 25%.

else

DISCOUNT is 15%.

- Structured english
 - describe business logic
- Pseudo-code
 - Lang agnostic generic notation for actual code
- Capitalised work in data dictionary

1.3.4.3.3 Decision Tables

- Show logical structure
- Show all outcomes of a process
- Used to describe a logical process
- Includes:
 - Title
 - Conditions
 - Either have a Yes (Y) or No (N) answer.
 - Actions
 - Steps taken as a result of condition
 - If action is to be taken an x is placed in table, if blank, no action taken

Rules - a combo of

- An entry for each condition
- The resulting actions
- Each rule forms a vertical column of squares
- Number of rules determine no of Conditions
 - 1 rule creates 2 conditions
 - 2 rules = 4 and so on

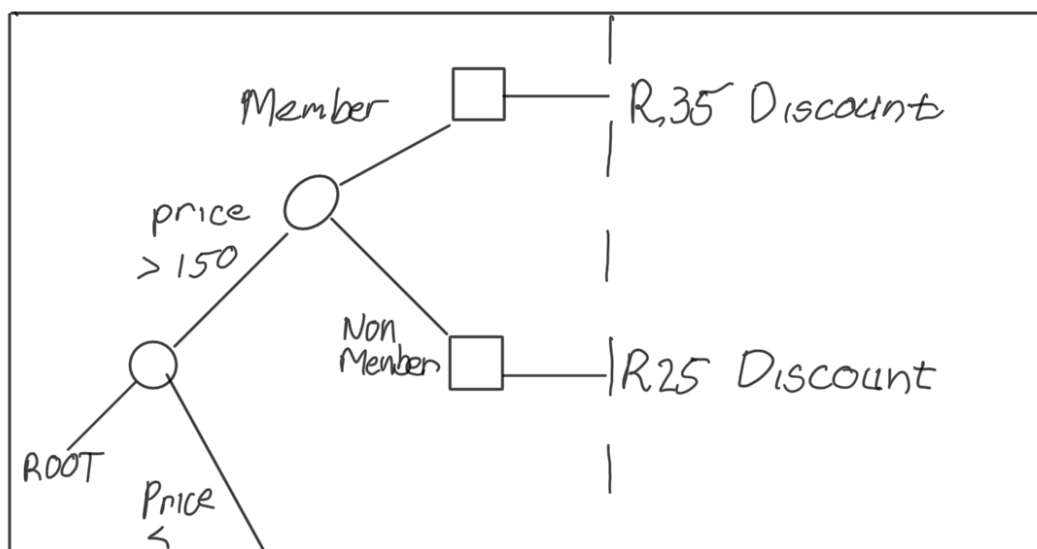
Price	Member	Discount
Y	Y	25%
Y	N	15%
N	Y	15%
N	N	15%

TABLE SIZE		CALCULATE DISCOUNT				TITLE SIZE OF TABLE
Stub		Entries				
		1	2	3	4	Rule header
C1	PRICE > 150	Y	Y	N	N	
C2	MEMBER	Y	N	Y	N	Rules
A1	R35 off	X				
A2	R25		X			
A3	NONE			X	X	

- Impossible rule
 - rule combo not possible

1.3.4.3 Decision tree

- Graphical representation of decision table
 - rules
 - conditions
 - actions
 - Shows logic horizontally
 - Read from left to right
 - Conditions appear along branches
 - Actions far right
 - Good for presenting sys to management





1.3.4.3.5 Logical & physical models

- Used to develop logical & physical models
 - Physical Model
 - How sys reqs implemented
 - follows from logical model
 - Developed during design phase
 - Operational tasks & techniques

1.3.4.3.6 Sequence Models

Analysis phase

- Before logical model
 - Understand system & operations involved
 - Create physical & logical models of current sys
 - Then create model of new system

1.3.4.3.7 4 Model Approach

- Develop logic and physical diagram
 - Old & new sys
 - Costly, time consuming
 - Better sys understanding
 - Mistakes avoided early

1.3.5 Object modelling

- O-O analysis
- Used to construct a model of sys requirements

1.3.5.1 O-O concepts

1.3.5.1 O-O concepts

Info Sys defined as objects
Represent real world things
Deliverable is object model
Represent entire info sys using o-o concepts
Objects transformed into module of code
during implementation phase
Modules easily modified & tested

LIML

- Presents IS graphically
- Develop object models
- Object
 - Includes data and processes (methods)
- DFDs
 - Separate data and process
- Attributes desc objects
eg Name etc
Object carries out task when instructed

1.3.5.1.1 Objects

- Shows as rectangle ÷ 3 parts



1.3.5.1.2 Attributes

- descriptive chars of an object
- Can be inherited from parents

1.3.5.1.3 Methods

- Define tasks object can do & how
- If method creates new object it's called Constructor
- Mutator method
 - Makes changes to existing data

- Accessor Method
 - Provide info about object attributes

1.3.5.1.4 Messages

- Tell objects to perform method
- Polymorphism
 - A message can mean diff things to diff objects
 - Blackbox
 - Message will tell object to do func not how
 - Known as encapsulation
 - All data & methods in object, nothing external needed

1.3.5.1.5 Classes

Objects are part of the class category

- If object belongs to a class
 - Likely other objects in the class
 - Share attributes & methods
- If there are specific categories in a class, subclasses are created
- Subclasses inherit attributes & methods from parents and have a few of their own
- Subclasses can have their own subclass
- When a class belongs to a more general class it's said to belong to super class

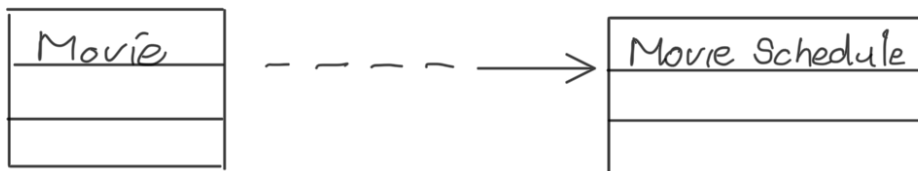
1.3.5.2 Relationships between classes & objects

- Objects have relationships
 - Interact this way
 - Help communicate

- Describe info objects need to know about others
- Show how changes in 1 obj affects another
- Show membership between classes, subclasses and superclasses
- Relationships can be very weak → Strong

1.3.5.2.1 Dependancy

- Weaker relationship
- 1 obj depends on another for info
- Changes in parent obj affect child
Movie depends on scheduled (objects)



1.3.5.2.2 Association

- Stronger
- If 1 object changes so does the other
 - Inheritance doesn't matter.
- eg Library object changes availability on book object

For more check doc 2 . . .

