MONASH University

FIT1013 Digital Futures: IT for Business
Week 9 : Modularisation, Structure Charts, Connecting to External Data

On completion of your study this week, you should aim to:

- Discuss program design approaches
- Design modules using structure charts
- Import data from external files

GROUP OF EIGHT AUSTRALIA

Two Approaches to System Development

MONASH University

- Traditional Approach
  - Also called structured system development
  - Structured analysis and design technique (SADT)
- Structured programming
  - Improves computer program quality
  - Allows other programmers to easily read and modify code
  - Each program module has one beginning and one ending

More details about Systems Development will be covered in **FIT2001**

Structured Analysis

MONASH University

- Define what system needs to do (processing requirements)
- Define data system needs to store and use (data requirements)
- Define inputs and outputs
- Define how functions work together to accomplish tasks
- Data flow diagrams (DFD) and entity relationship diagrams (ERD) show results of structured analysis

Structured Design

MONASH University

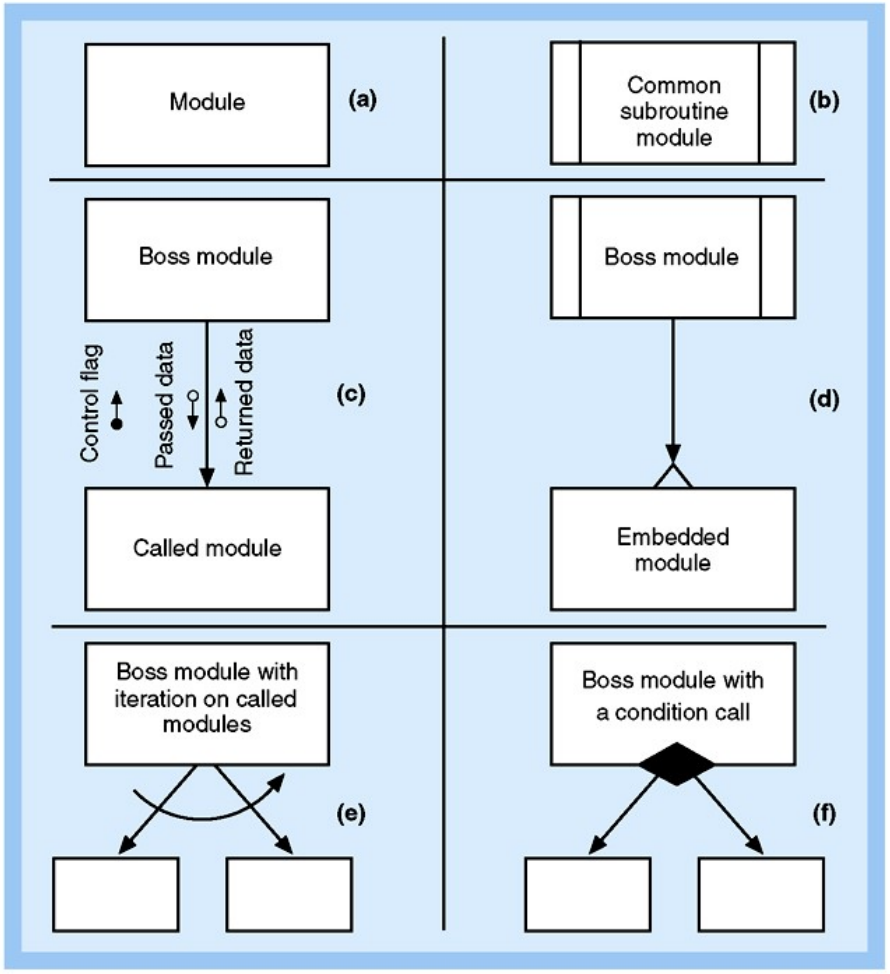
- Technique developed to provide design guidelines
  - What set of programs should be
  - What program should accomplish
  - How programs should be organized into a hierarchy
- Modules are shown with structure chart
- Main principle of program modules
  - Loosely coupled – module is independent of other modules
  - Highly cohesive – module has one clear task

The Structure Chart

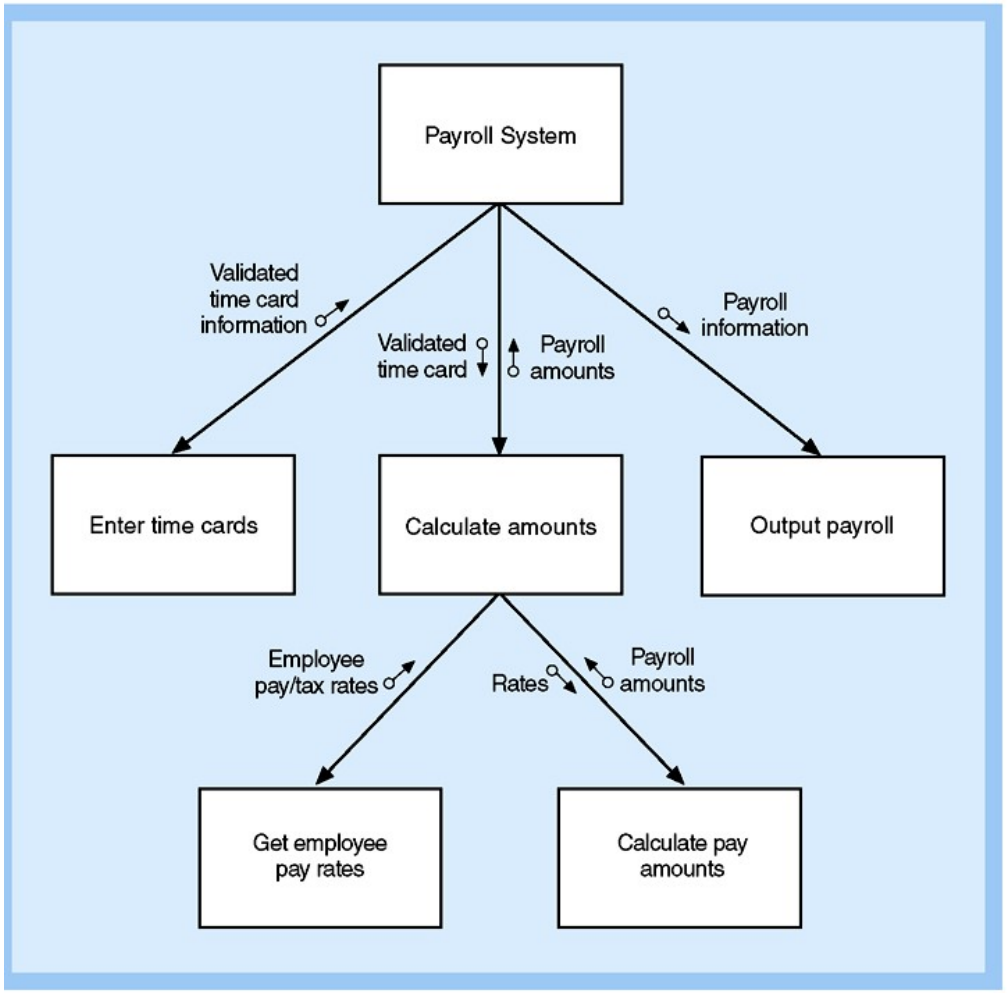
MONASH University

- Describes functions and sub functions of each part of system
- Shows relationships between modules of a computer program
- Simple and direct organization
  - Each module performs a specific function
  - Each layer in a program performs specific activities
- Chart is tree-like with root module and branches

Structure Chart Symbols

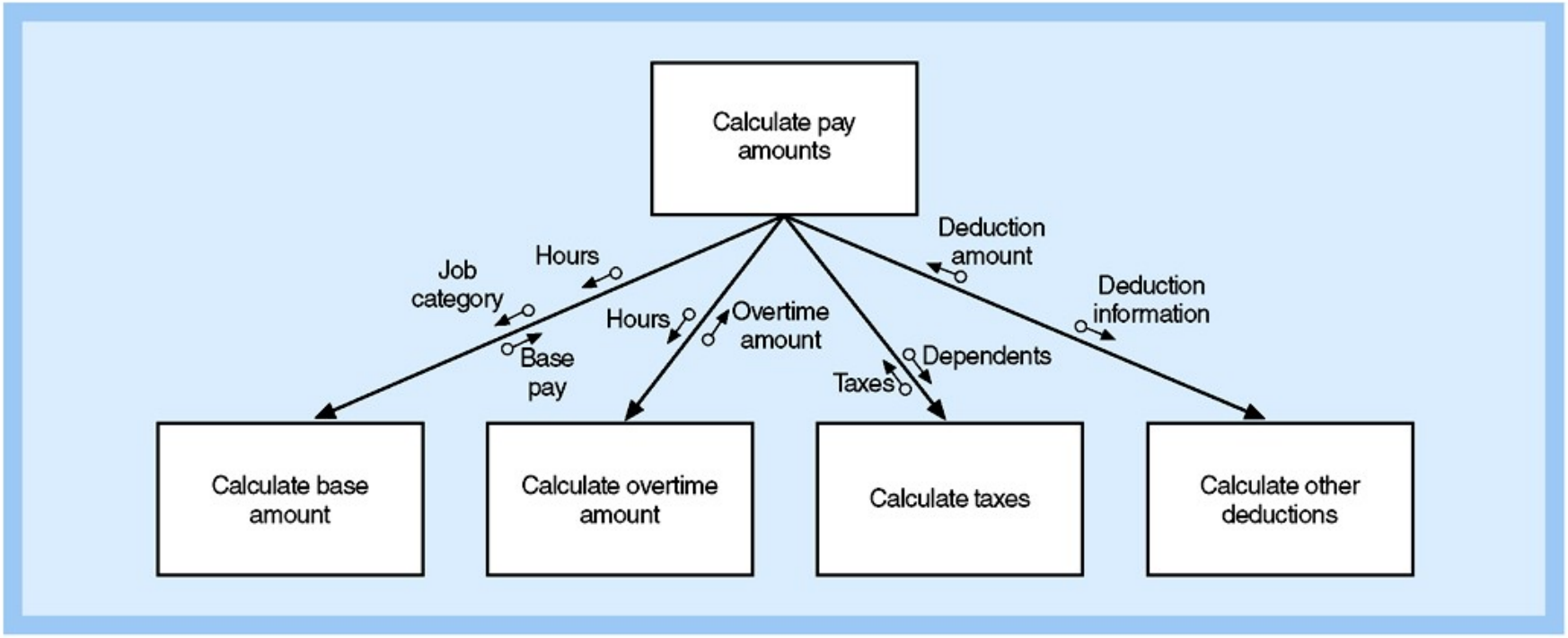


Structure Chart Created Using Structured Design Technique

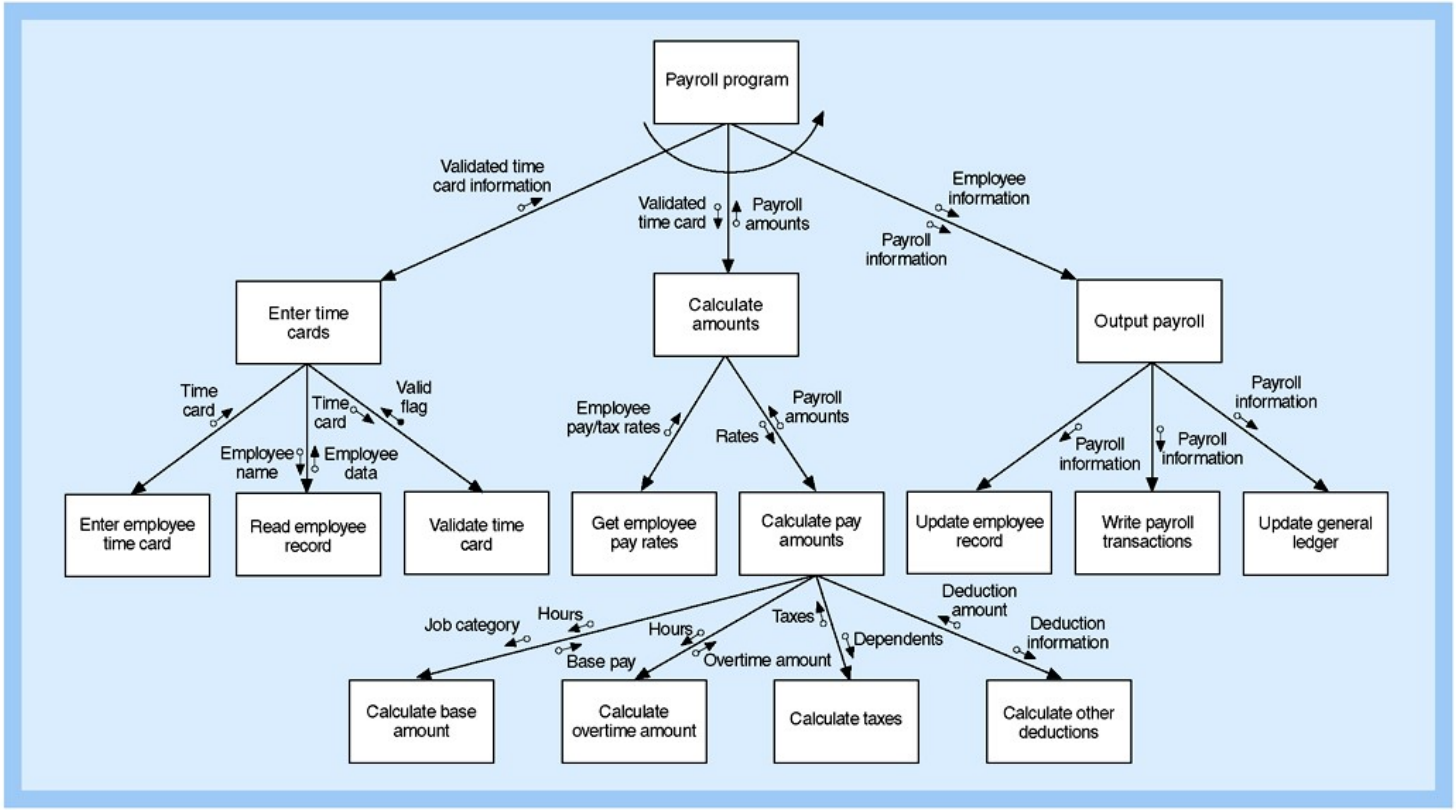


Example of a structure chart

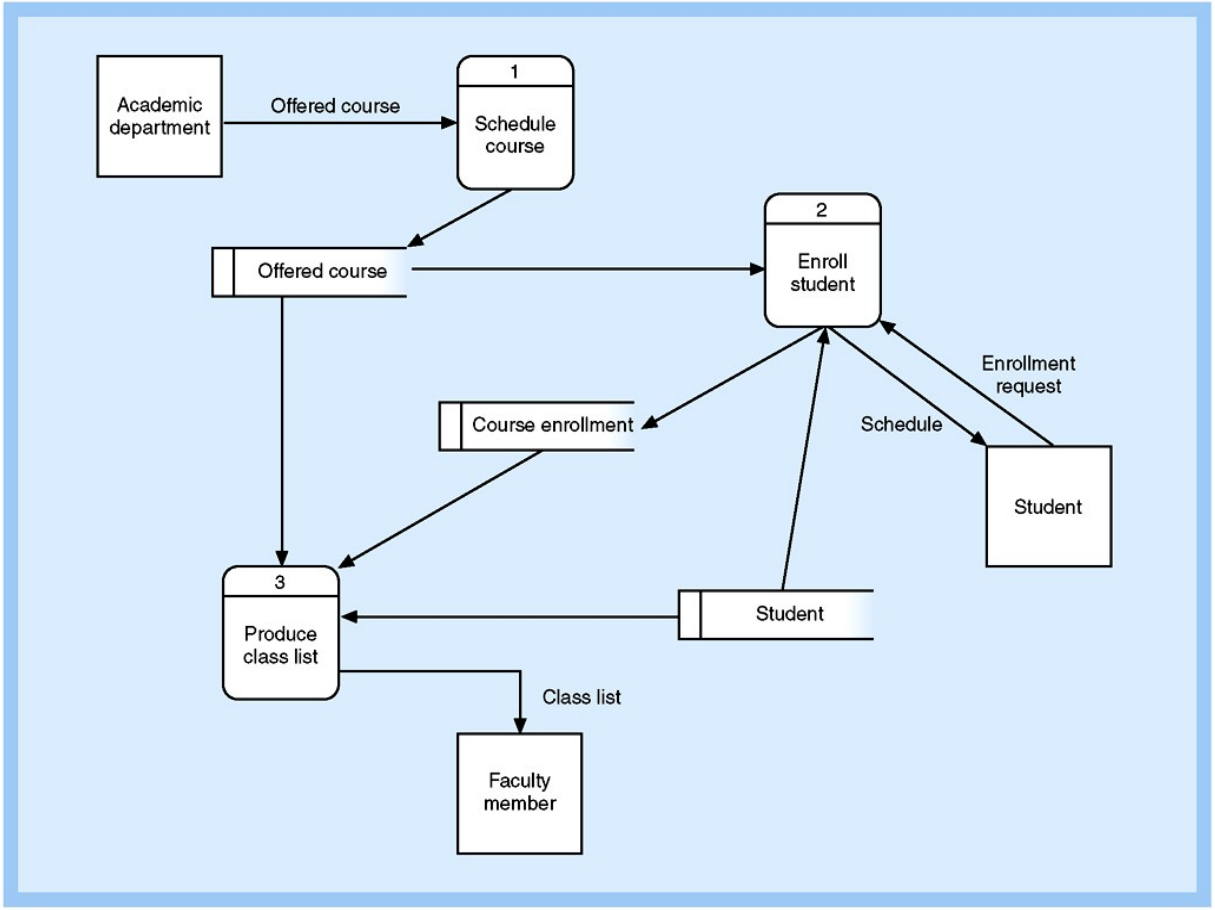
A Simple Structure Chart for the Calculate Pay Amounts Module



Structure Chart for Entire Payroll Program

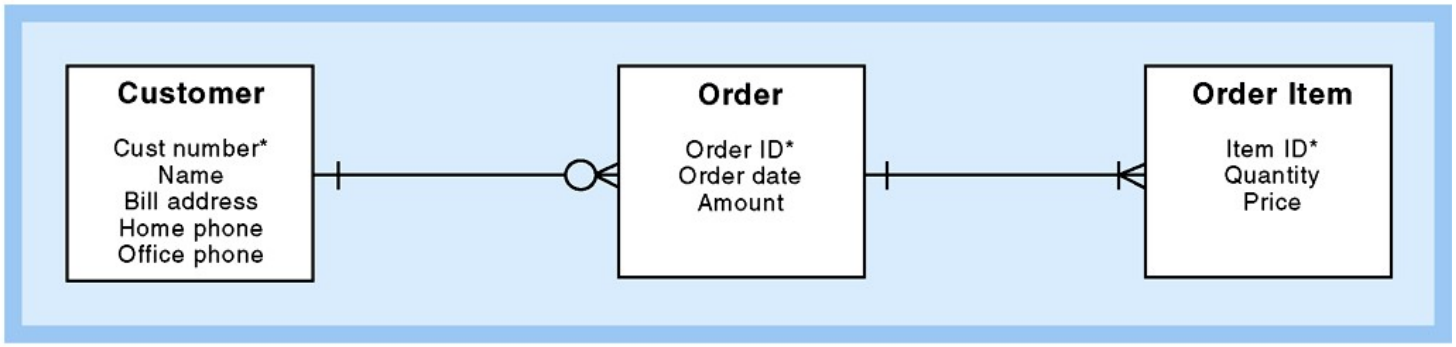


Data Flow Diagram (DFD) created using Structured Analysis Technique



Note: **FIT2090** will cover the topic on DFD

Entity-Relationship Diagram (ERD) created using the Structured Analysis technique



Note: ERD is covered in the **database units**

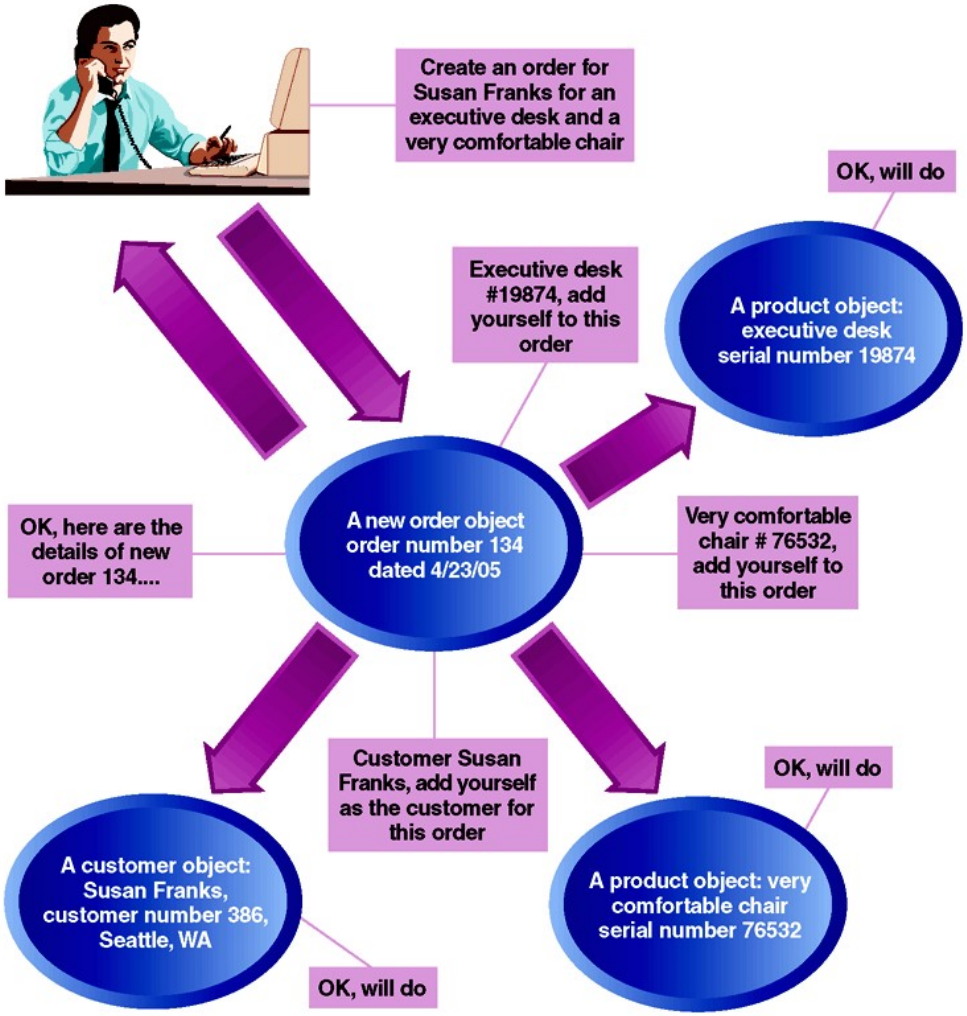
Object-Oriented Approach

MONASH University

- Views information system as collection of interacting objects that work together to accomplish tasks
  - Objects - things in computer system that can respond to messages
  - No processes, programs, data entities, or files are defined – just objects
- Object-oriented analysis (OOA)
  - Defines types of objects that do work of system
  - Shows how objects interact with users to complete tasks

Object-Oriented Approach to Systems

MONASH University



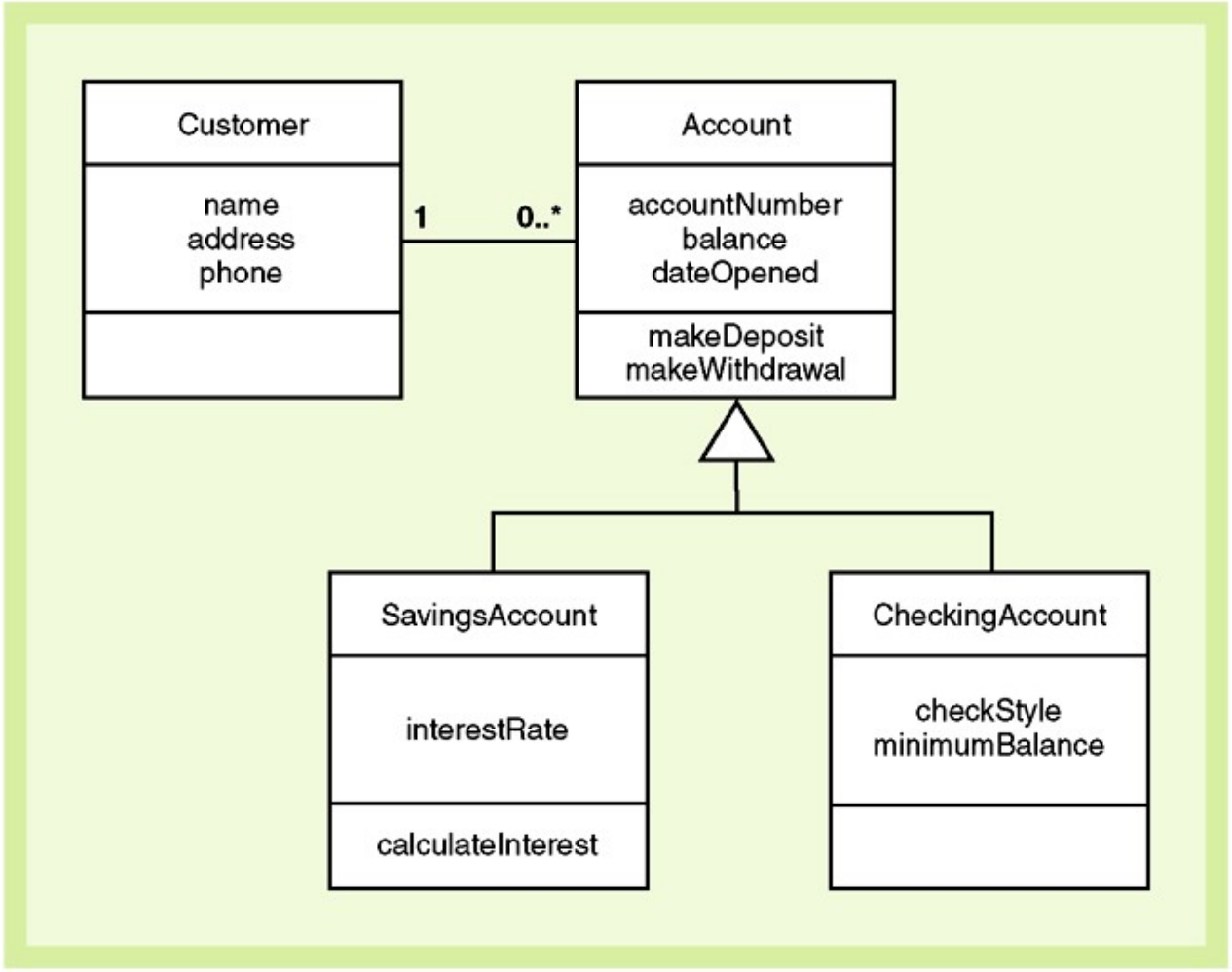
Object-Oriented Approach (continued)

MONASH University

- Object-oriented design (OOD)
  - Defines object types needed to communicate with people and devices in system
  - Shows how objects interact to complete tasks
  - Refines each type of object for implementation with specific language of environment
- Object-oriented programming (OOP)
  - Writing statements in programming language to define what each type of object does
- Benefits of OOA include naturalness and reuse

Class Diagram Created During OO Analysis

MONASH University



Developing a Structure Chart

MONASH University

- Transaction Analysis
  - Uses system flow chart and event table inputs
  - Upper-level modules developed first
  - Identifies each transaction supported by program
- Transform Analysis
  - Uses **DFD** fragments for inputs
  - Computer program ‘transforms’ inputs into outputs
  - Charts have input, calculate, and output subtrees

Steps to Create a Structure Chart from a DFD Fragment

MONASH University

- Determine primary information flow
  - Main stream of data transformed from some input form to output form
- Find process that represents most fundamental change from input to output
- Redraw **DFD** with inputs to left and outputs to right – central transform process goes in middle
- Generate first draft structure chart based on redrawn data flow

Data Flow Diagrams

MONASH University

- Data flow diagram: a graphical representation of a system that depicts the systems components; the data flows among the components; and the sources, destinations, and storage of data.
- Use a limited number of symbols.
- Do not depict management or operational elements of a system.

What are DFDs used for?

MONASH University

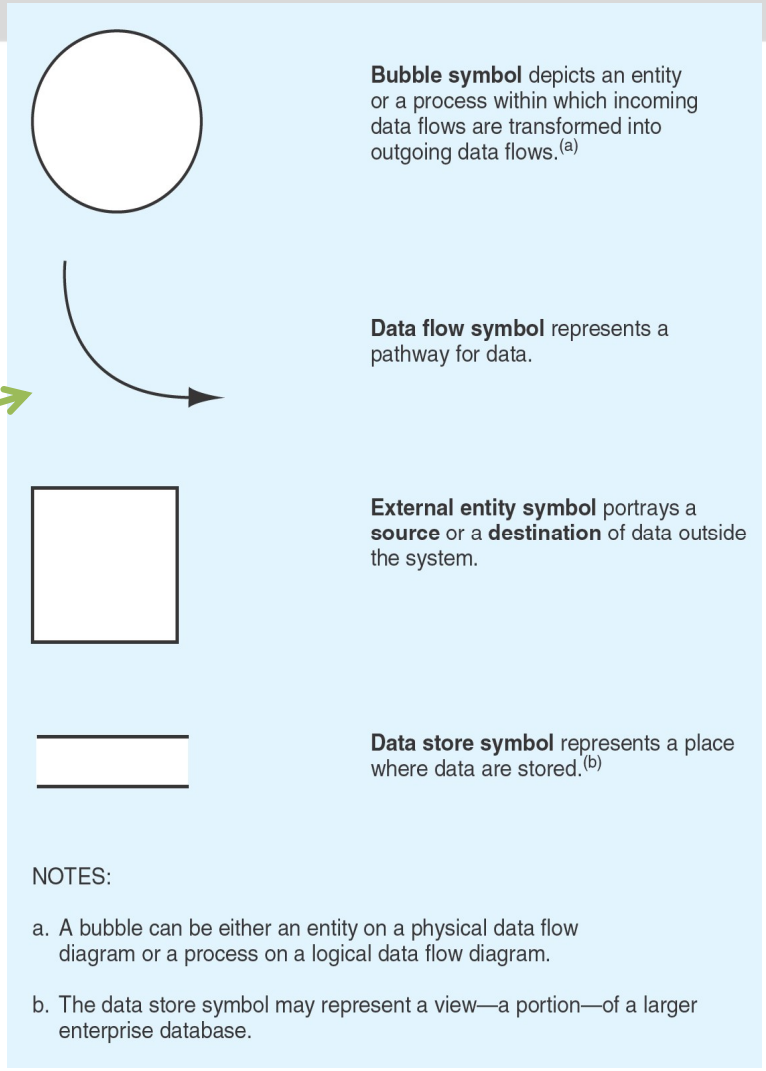
- A new (or part of an) implementation of a system
- A new (or part of a) **business process reengineering (BPR)** project
- As part of the assessment of internal controls (e.g. for auditing purposes)

Note: BPR will be discussed in **FIT2090**

Basic DFD Symbols

MONASH University

- Movement of data among:
  - Entities (sources or destinations)
  - Processes
  - Data stores.
- Label should describe the information moving



Data Flow Diagram Levels

MONASH University

- Context
  - Highest level (most general)
  - Purpose: show inputs and outputs into system
  - Characteristics: one process symbol only, no data stores.
- Level-0
  - Purpose: show all major activity steps of a system
  - Characteristics: processes are labeled 1.0, 2.0 and so on.
- Level-1
  - Purpose: show one major activity divided into sub-activities
  - Characteristics: processes are labeled 1.1, 1.2 and so on.

More details are covered in FIT2090

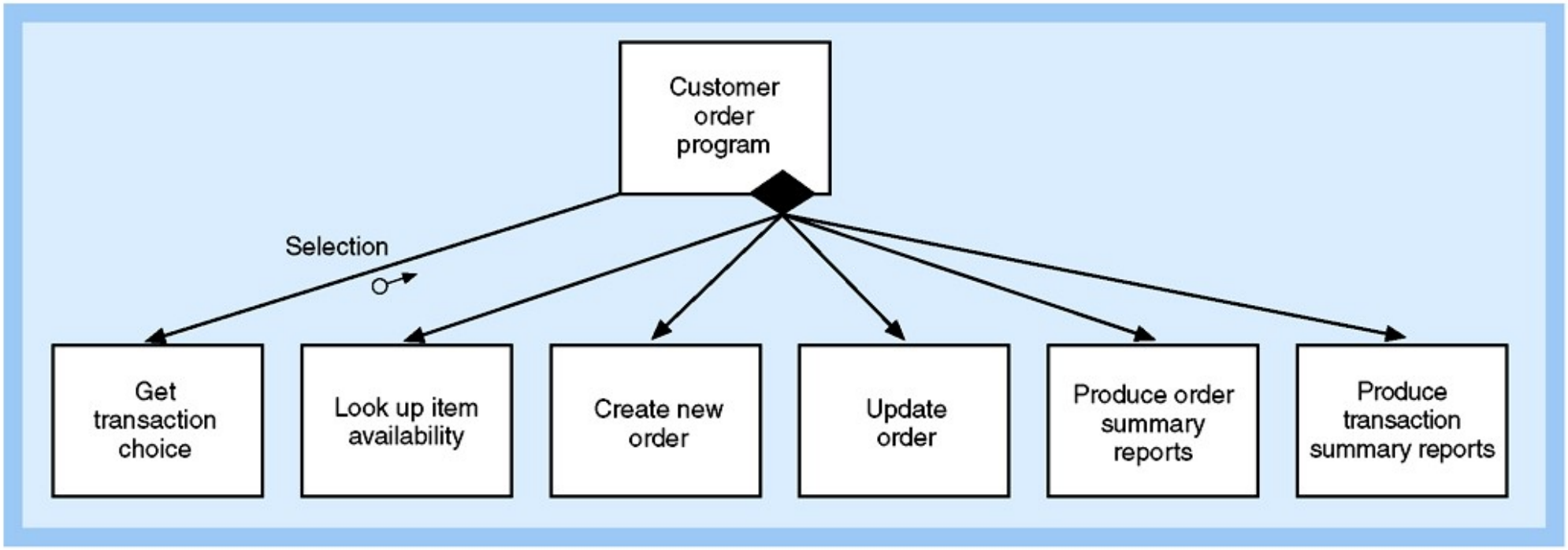
Context Diagram

MONASH University

- Context diagram
  - top-level, and less detailed, diagram of a system depicting the system and all its activities as a single bubble and showing the data flows into and out of the system and into and out of the external entities.
- External entities
  - those entities (i.e., persons, places, or things) outside the system that send data to, or receive data from, the system.
- Internal entities
  - those entities within the system that transform data
  - Includes, for example, accounting clerks (persons), departments (places), and computers (things)

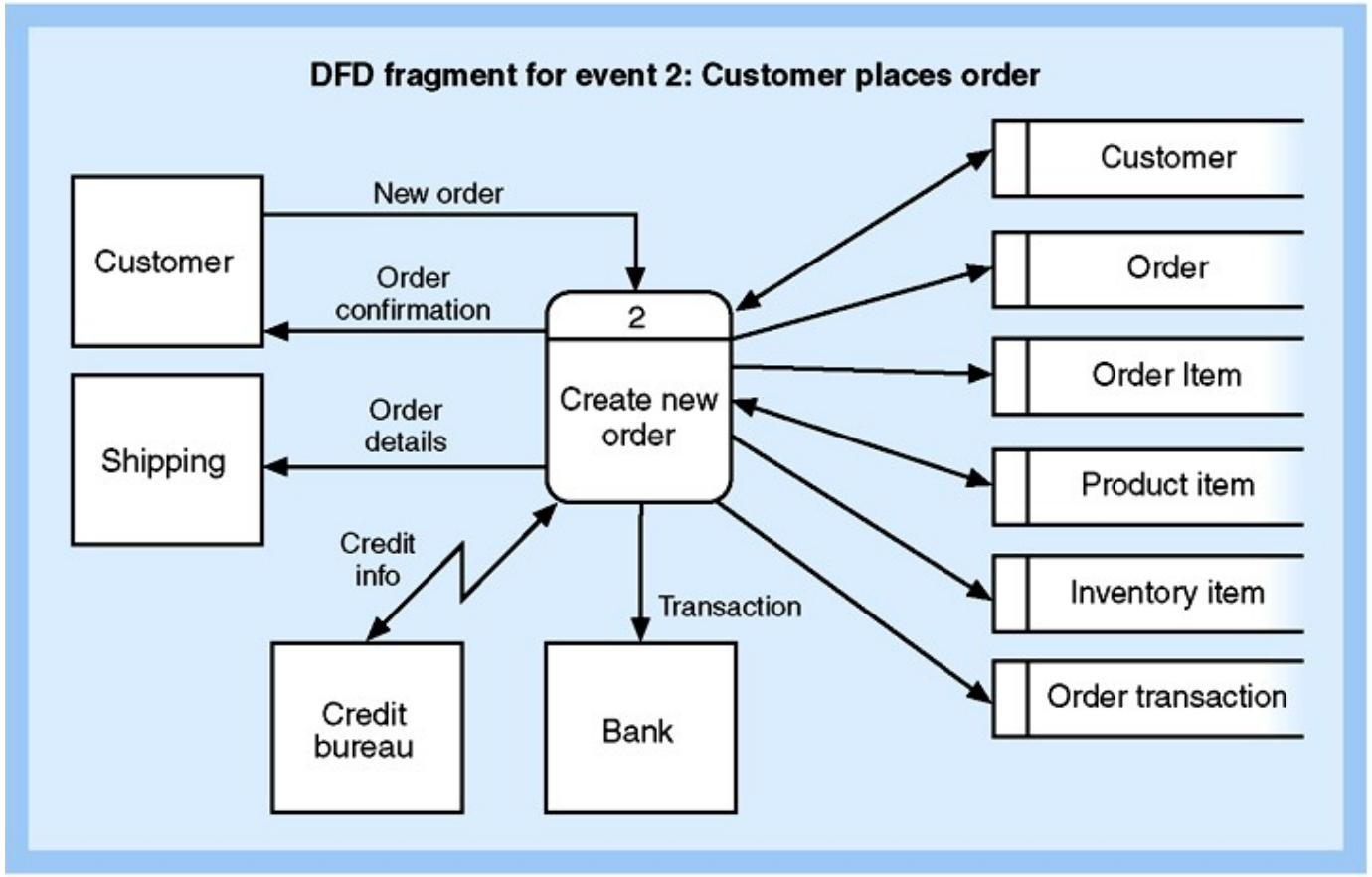
High-level Structure Chart for the Customer Order Program

MONASH University

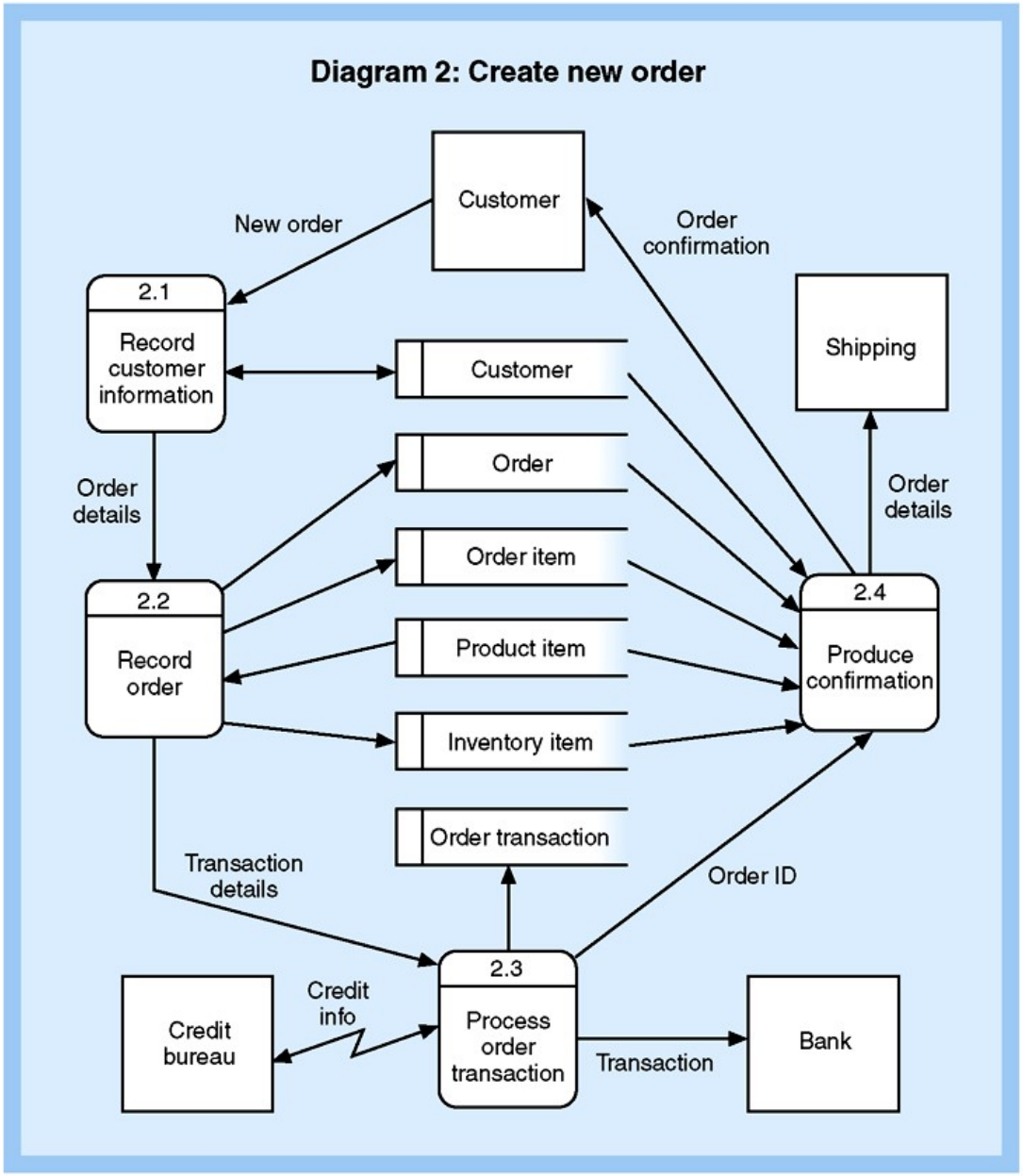


The *Create New Order* DFD Fragment

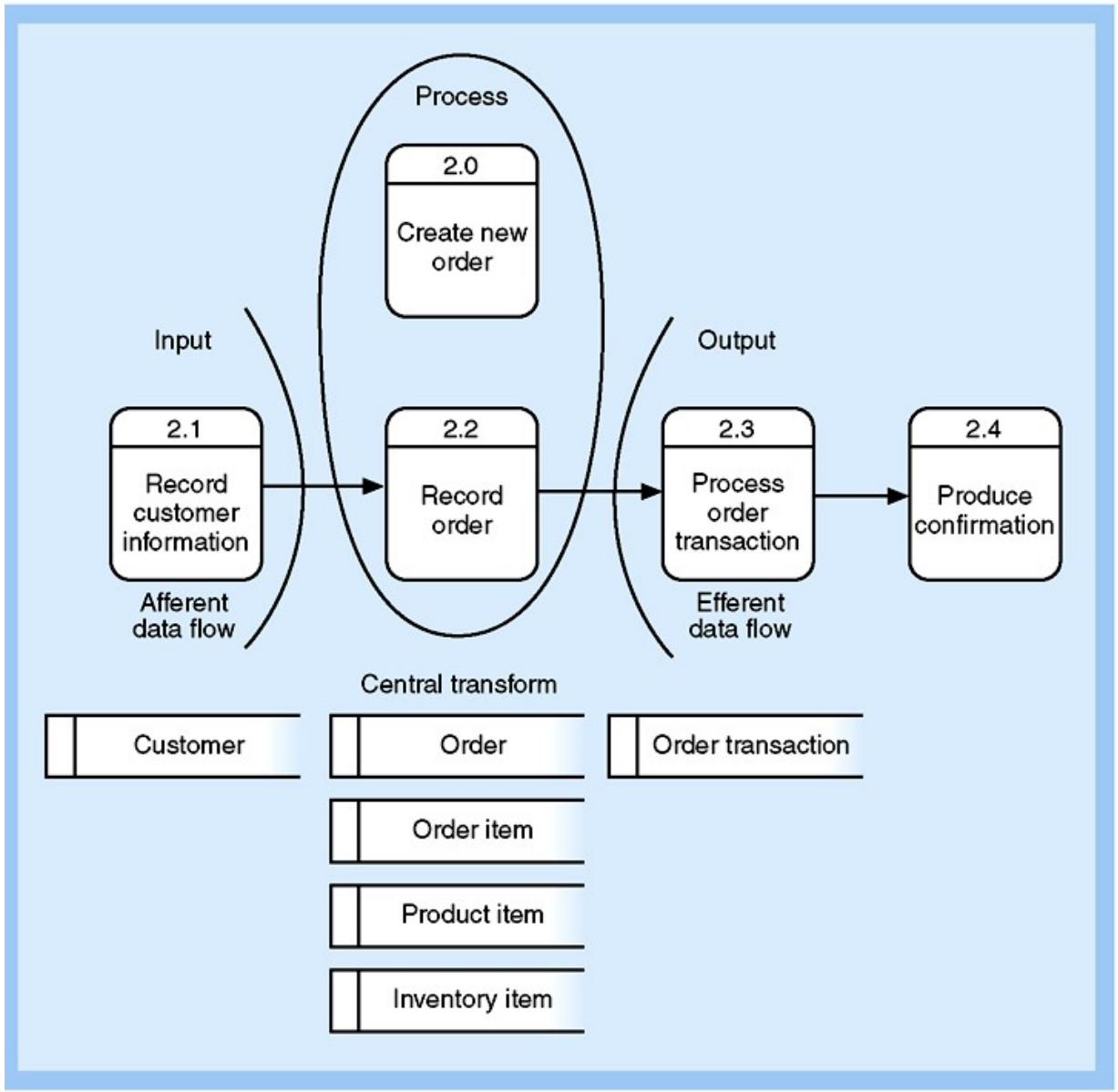
MONASH University



Exploded View of Create New Order DFD

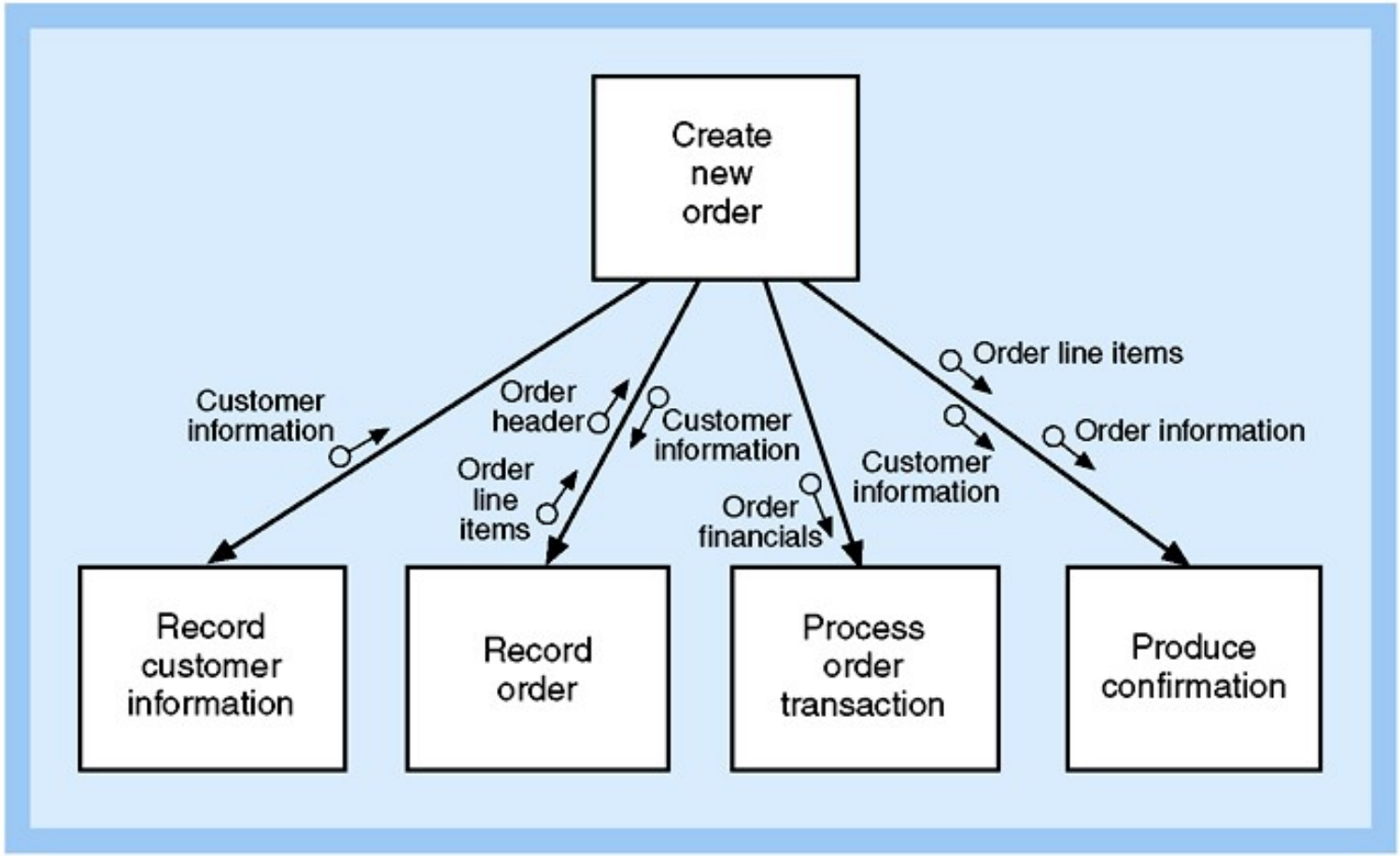


Rearranged Create New Order DFD



First Draft of the Structure Chart

MONASH University



Steps to Create a Structure Chart from a DFD Fragment (continued)

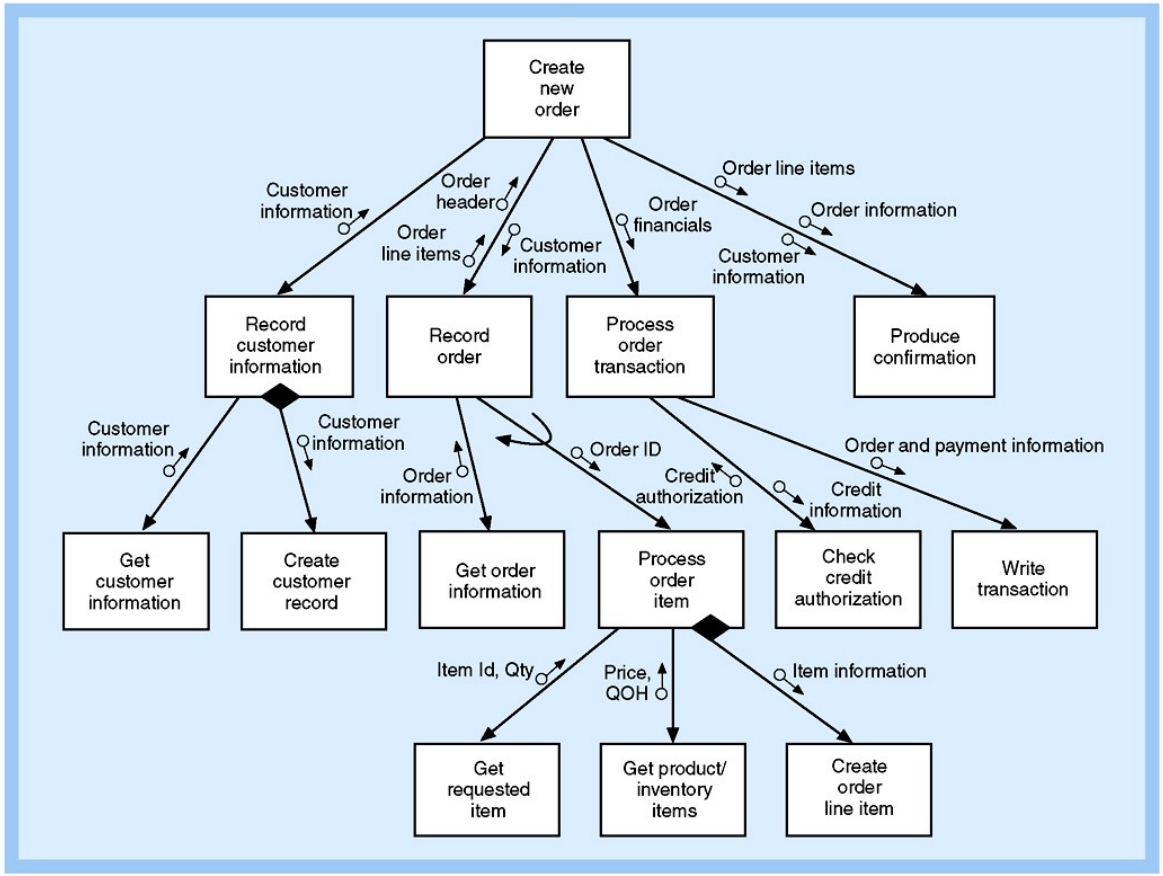
MONASH University

- Add other modules
  - Get input data via user-interface screens
  - Read from and write to data storage
  - Write output data or reports
- Add logic from structured English or decision tables
- Make final refinements to structure chart based on quality control concepts

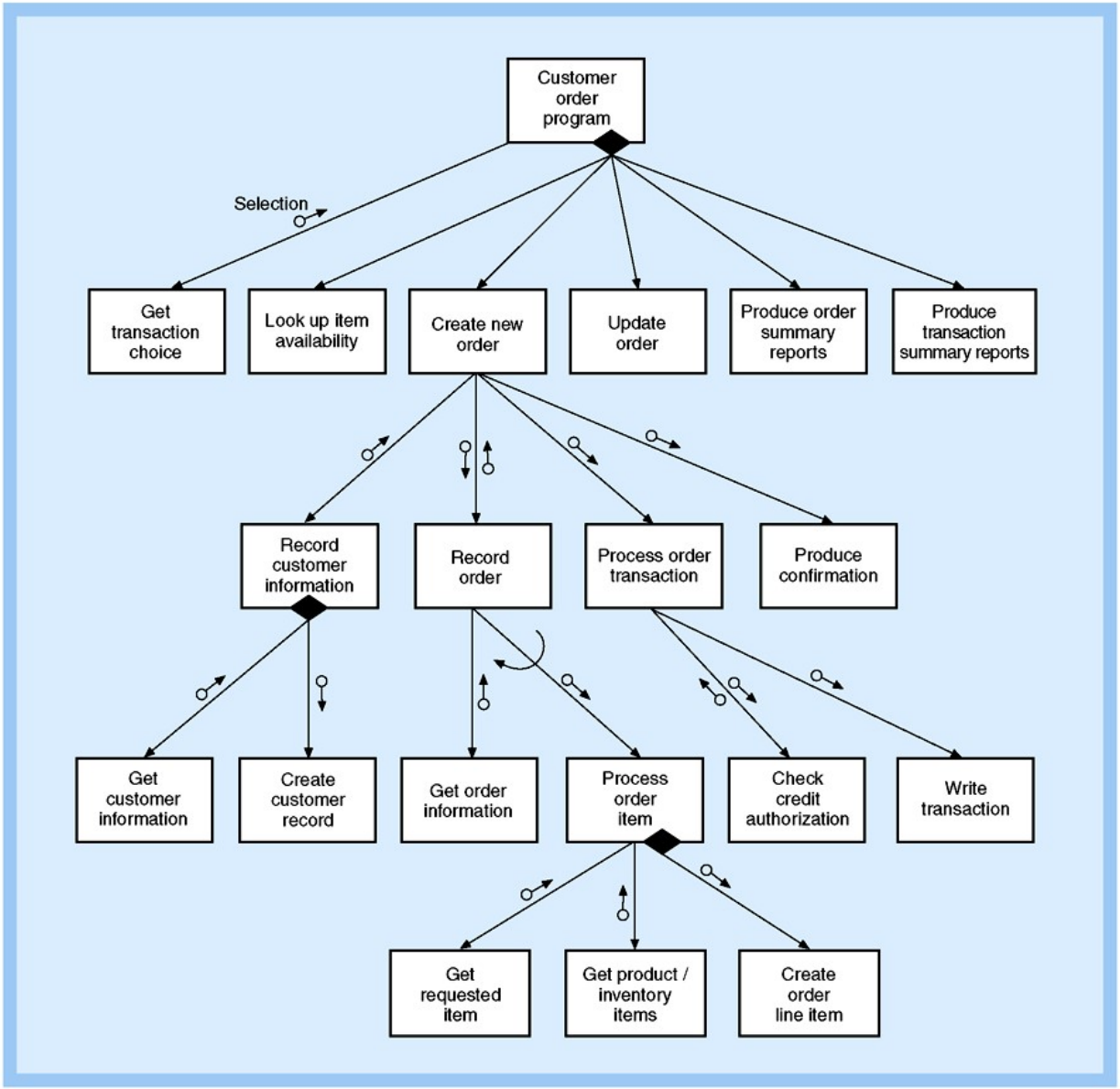
The Structure Chart for the *Create New Order* Program

FIGURE 10-15

The structure chart for the *Create new order* program.



Combination of Structure Charts

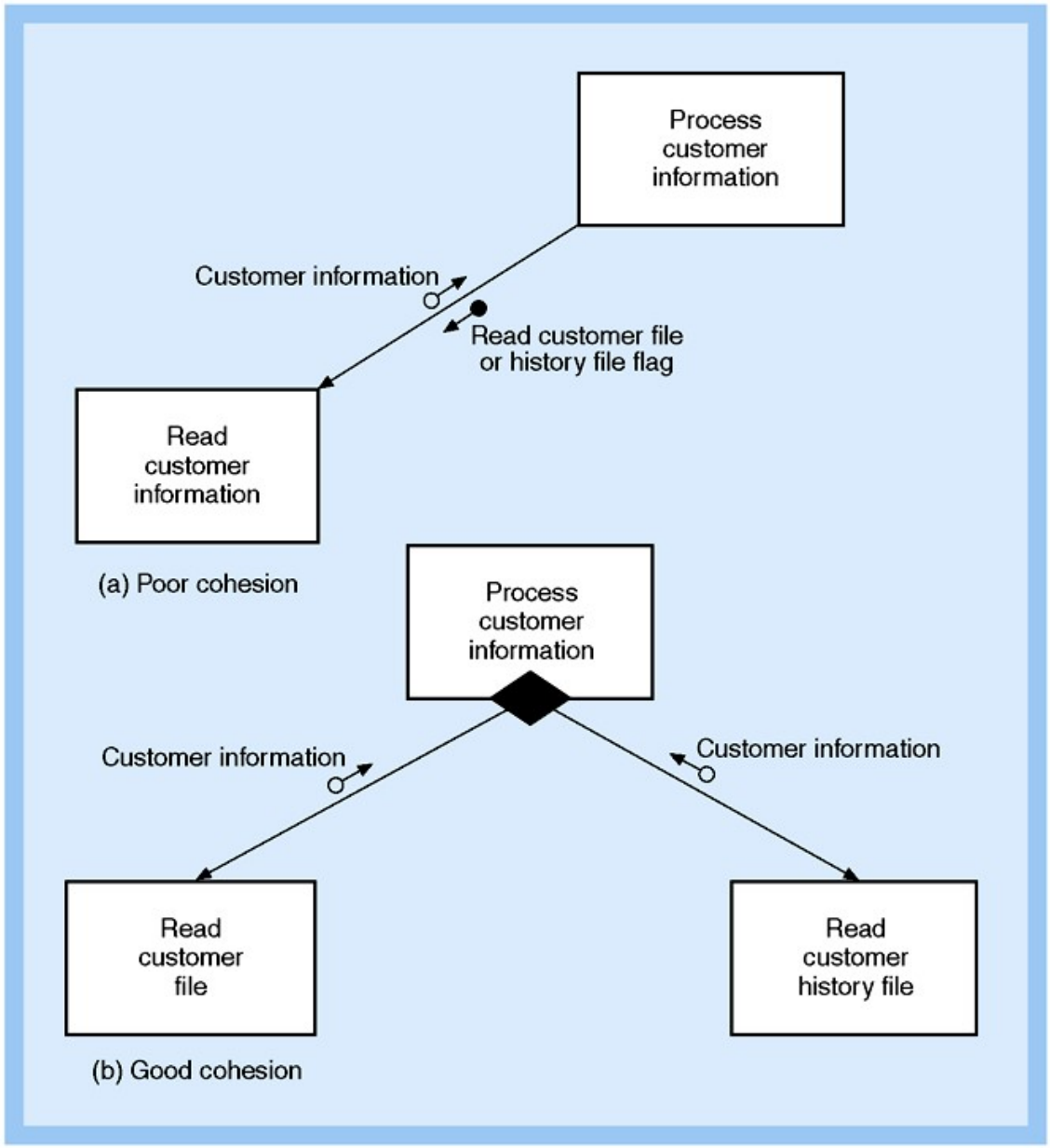


Evaluating the Quality of a Structure Chart

MONASH University

- Module coupling
  - Measure of how module is connected to other modules in program
  - Goal is to be loosely coupled
- Module cohesion
  - Measure of internal strength of module
  - Module performs one defined task
  - Goal is to be highly cohesive

Examples of Module Cohesion



Module Algorithm Design: Pseudocode

MONASH University

- Describes internal logic of software modules
- Variation of structured English that is closer to programming code
- Syntax should mirror development language
- Three types of control statements used in structured programming:
  - Sequence: sequence of executable statements
  - Decision: if-then-else logic
  - Iteration: do-until or do-while

Integrating Structured Application Design with Other Design Tasks

MONASH University

- Structure chart must be modified or enhanced to integrate design of user interface and database
  - Are additional modules needed?
  - Does pseudocode in modules need modification?
  - Are additional data couples needed to pass data?
- Structure charts and system flowcharts must correspond to planned network architecture
  - Required protocols, capacity, and security

Structure Chart Showing the Create New Order Program