Information Systems (තොරතුරු පද්ධති)

Lesson 05 – Part 01

(System Analysis & Design)

Objectives – Learning Outcomes

The objectives of this section are to get knowledge on system analysis.

When you have followed this section you will:

- Understand to identify the Functional Requirements and Nonfunctional requirements
- Demonstrate how Activity diagrams and Document Flow Diagrams (Analytical Tools) work
- Understand to identify the concept of Logical Design

Outline

- Requirements
  - Functional Requirements / Nonfunctional requirements
- Analytical Tools
  - Activity diagrams / Document flow diagrams /Data flow diagrams
- Business system options
  - Logical Design
  - Architectural design / Logical data structures
- Process specification
- Data dictionary
- Interface design

What are Software Requirements ?

- The software requirements are description of features and functionalities of the target system.
- Requirements convey the expectations of users from the software product.
- The requirements can be obvious or hidden, known or unknown, expected or unexpected from client’s point of view.

Mainly there are two types,

- Functional Requirement
- Non-Functional Requirements

Functional Requirements

- Requirements, which are related to functional aspect of software fall into this category.
- They define functions and functionality within and from the software system.

- Ex – If it is Inventory Management System,
  - It should be able to provide reports
  - It should be able to add/delete/edit new entries

Non-Functional Requirements

- Requirements, which are not related to functional aspect of software, fall into this category.
- They are implicit or expected characteristics of software, which users make assumption of.

Non-functional requirements include -

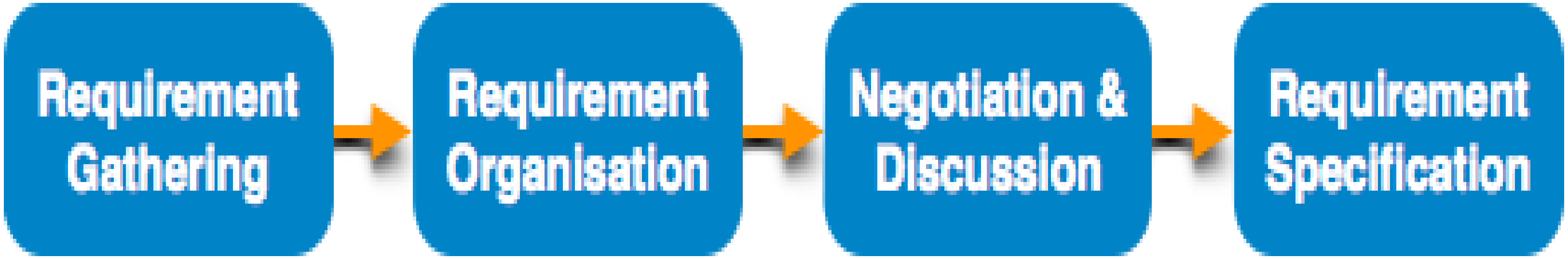
Security, Logging, Storage, Configuration
Performance, Cost, Interoperability
Flexibility, Disaster recovery, Accessibility

Requirement Engineering Process

This is a four step process,

- Feasibility Study
- Requirement Gathering
- Software Requirement Specification
- Software Requirement Validation

Requirement Elicitation Process



Requirement Gathering → Requirement Organisation → Negotiation & Discussion → Requirement Specification

Requirement Elicitation Techniques

- Interviews
- Surveys
- Questionnaires
- Task analysis
- Domain Analysis
- Brainstorming
- Prototyping
- Observation

System/SW Requirement Characteristics

- Clear
- Correct
- Consistent
- Coherent
- Modifiable
- Verifiable
- Prioritized
- Unambiguous
- Traceable
- Credible source
- Comprehensible

Must Have : Software cannot be said operational without them.

Should have : Enhancing the functionality of software.

Could have : Software can still properly function with these requirements.

Wish list : These requirements do not map to any objectives of software.

Analytical Tools

1. Activity Diagrams
2. Document Flow Diagrams
3. Data Flow Diagrams

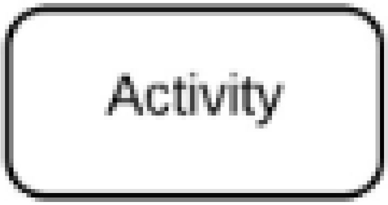
1. Activity Diagrams

- What is an Activity Diagram

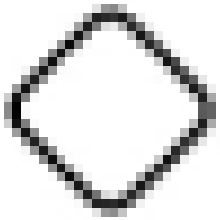
- Activity diagrams represent the dynamics of the system.
- They are flow charts that are used to show the workflow of a system.

Components of an Activity Diagram

- **Actions/Activity** - This is symbolized with a round-edged rectangle.

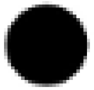


- **Decision node** - A conditional branch in the flow that is represented with a diamond. It includes a single input and two or more outputs.

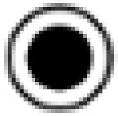


Components of an Activity Diagram

- **Start node** - Symbolizes the beginning of the activity. This is represented with a black circle.



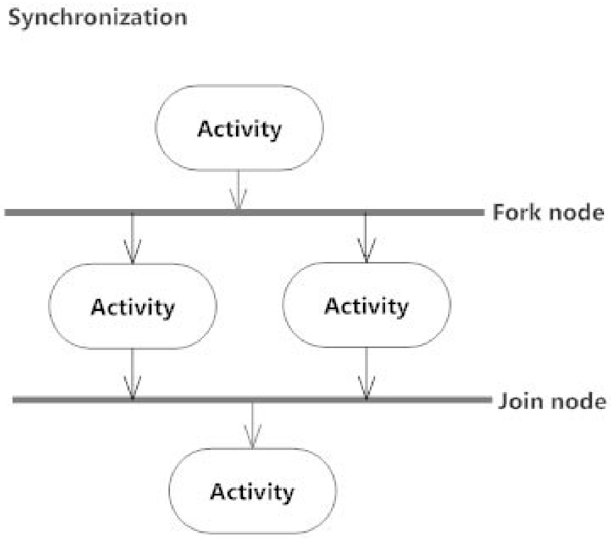
- **End node** - Represents the final step in the activity. It's modeled with an outlined black circle.



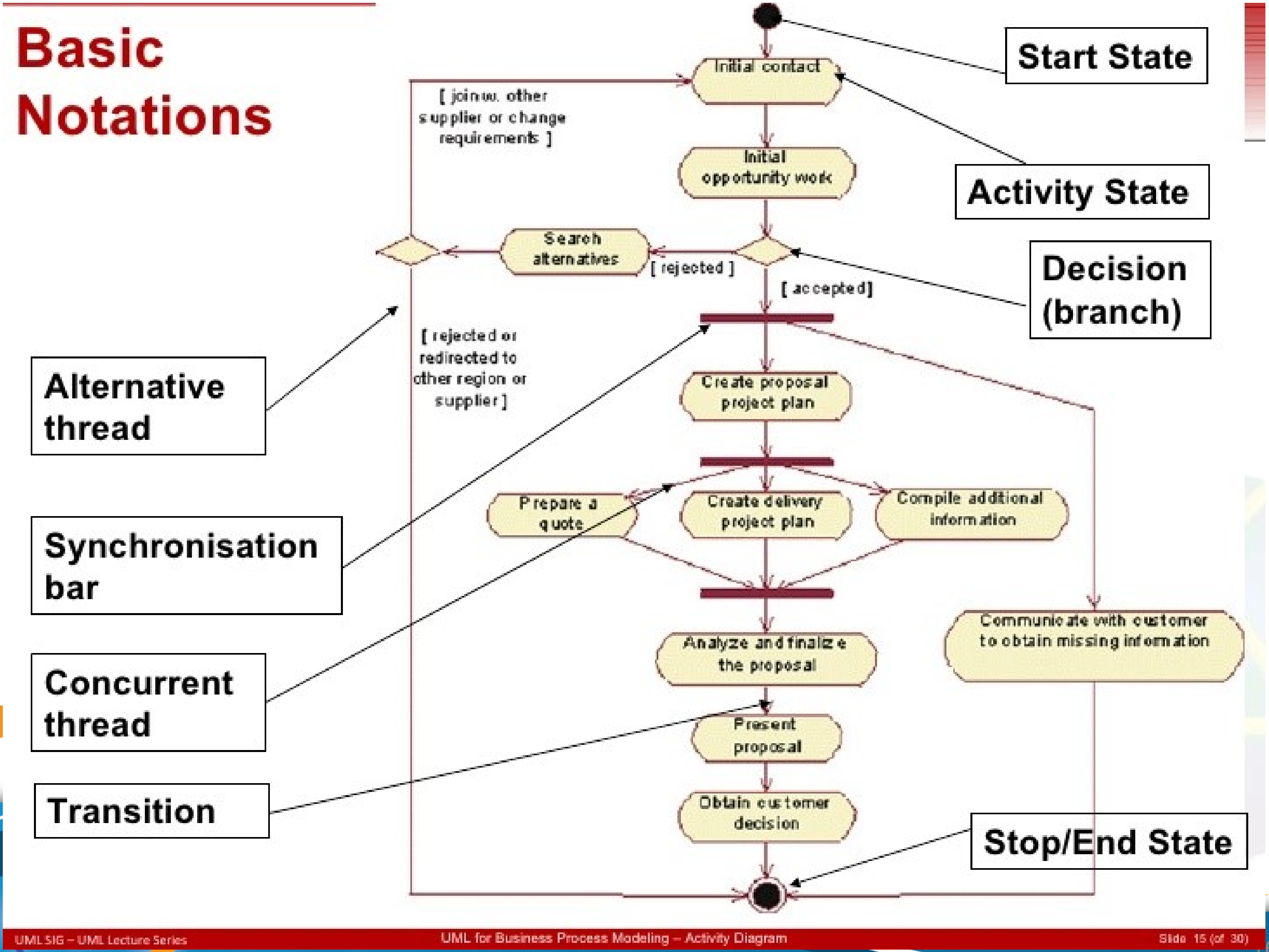
Components of an Activity Diagram

- **Control flows/Arrows** - Arrows represent the direction flow of the flow chart. The arrow points in the direction of progressing activities.

- **Synchronization Bar -** A fork node is used to split a single incoming flow into multiple concurrent flows.



Activity Diagrams



Basic Notations

Start State

Activity State

Decision (branch)

Alternative thread

Synchronisation bar

Concurrent thread

Transition

Stop/End State

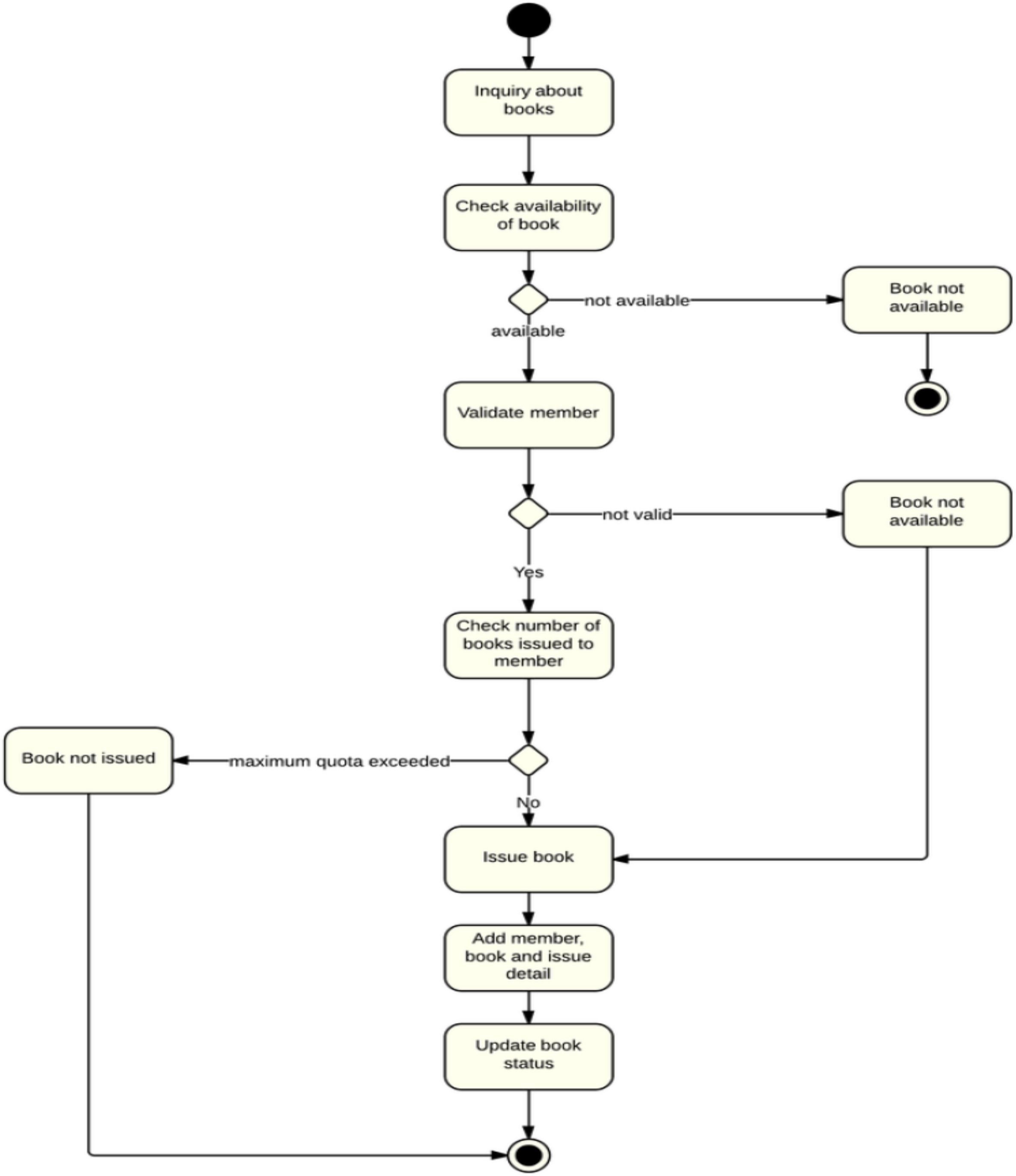
Initial contact
[join w. other supplier or change requirements]
Initial opportunity work
Search alternatives
[rejected]
[accepted]
[rejected or redirected to other region or supplier]
Create proposal project plan
Prepare a quote
Create delivery project plan
Compile additional information
Communicate with customer to obtain missing information
Analyze and finalize the proposal
Present proposal
Obtain customer decision

Activity Diagrams

Draw the activity diagram for below example

- Library Information System – (Functions)
  - Inquire about books
  - Check the availability of books
  - Member validation
  - Check the issued books to member
  - Add new members
  - Update the book status

Library Information System

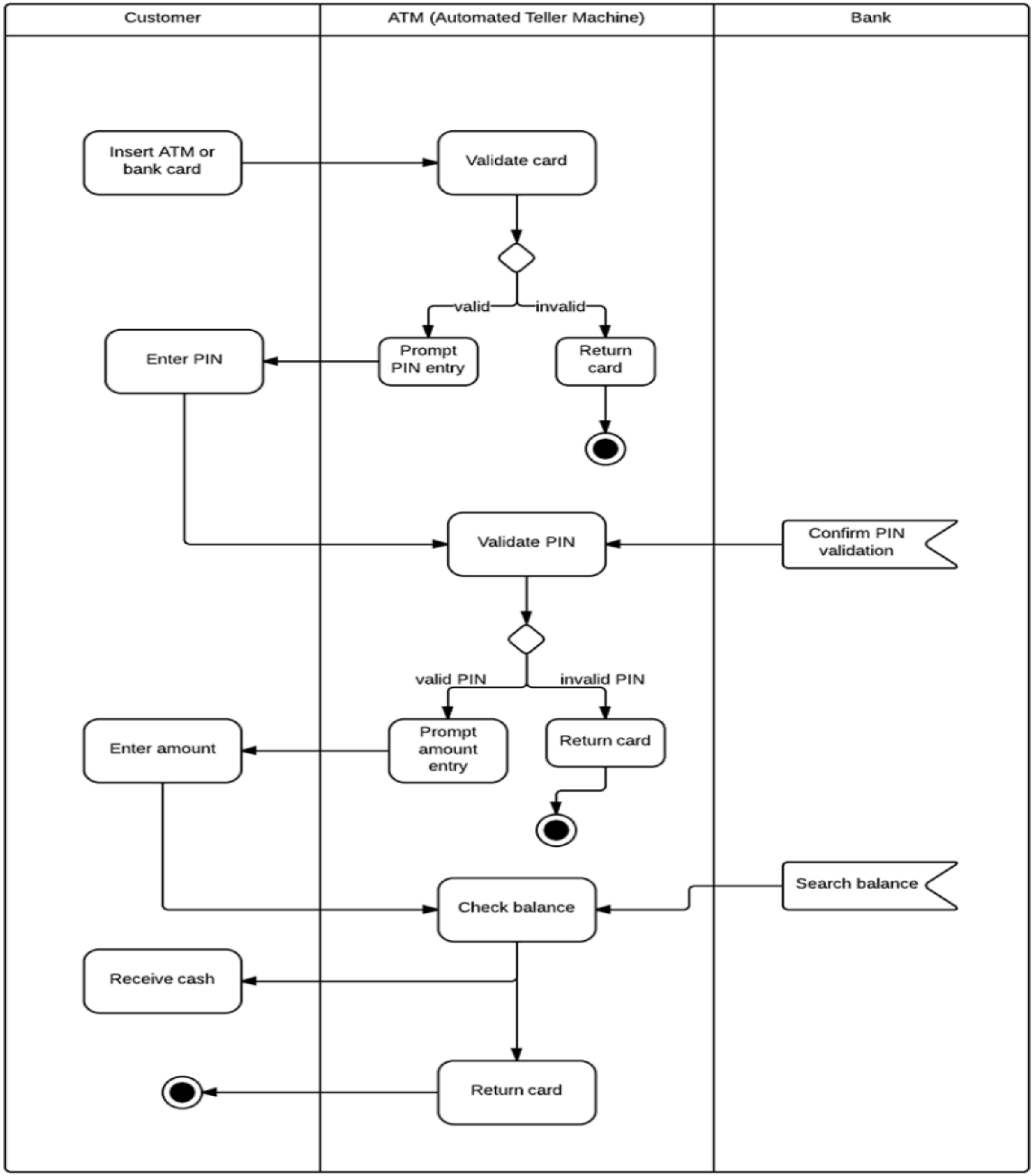


Activity Diagrams

Draw the activity diagram for below example

- Bank ATM System – (Functions)
  - Insert card
  - Enter pin
  - Enter amount
  - Receive cash
  - Validate card
  - Validate pin
  - Check balance
  - Return card
  - Confirm validation
  - Check balance

Bank ATM System

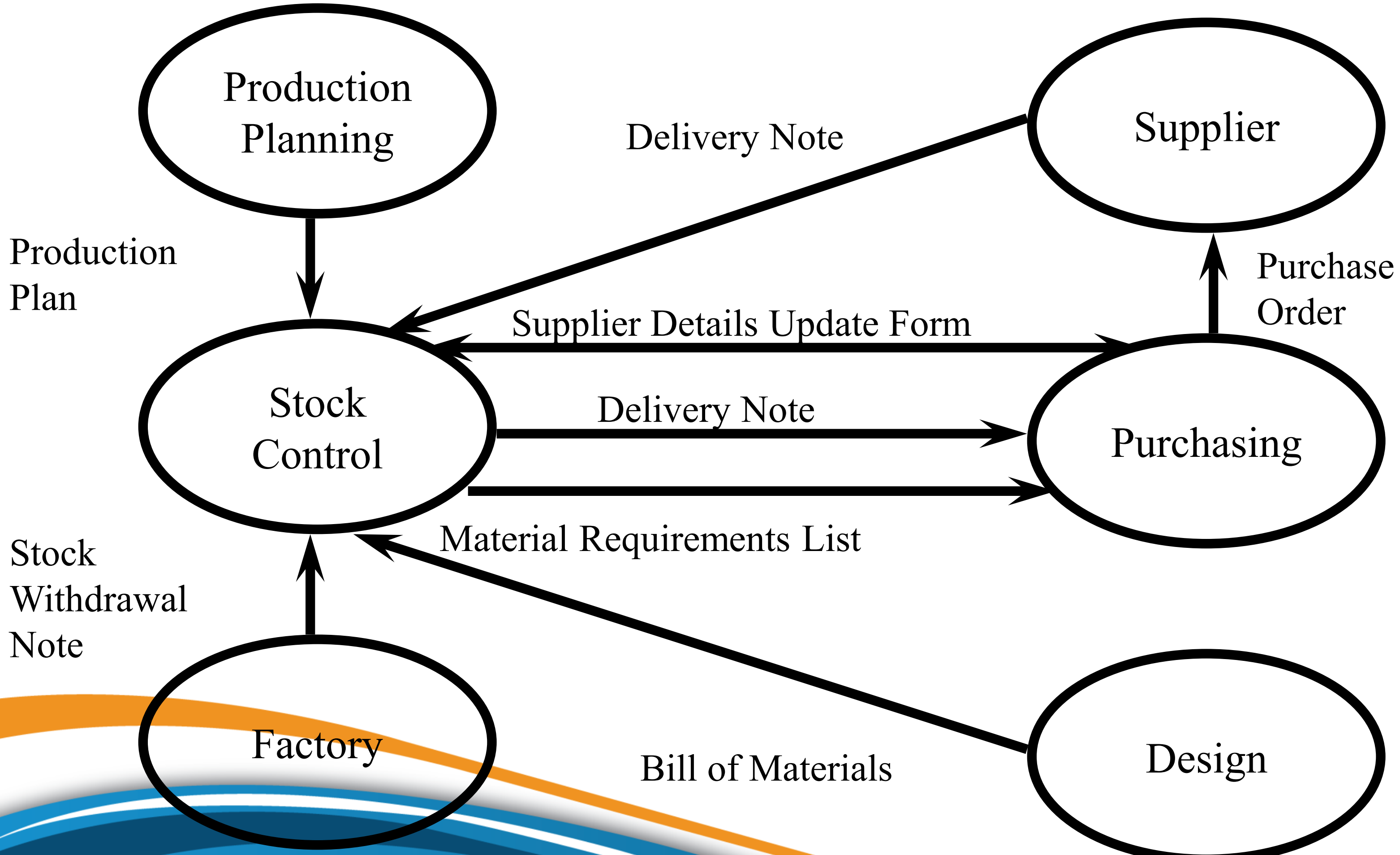


2. Document Flow Diagrams

- What is a Document Flow Diagram

- Document flow diagram is used to model the flow of documentation between external entities.

Document Flow Diagrams - example



3. Data Flow Diagrams

- What is a Data Flow Diagram

- Data-flow diagrams (DFDs) are system models that show a functional perspective where each transformation represents a single function or process.

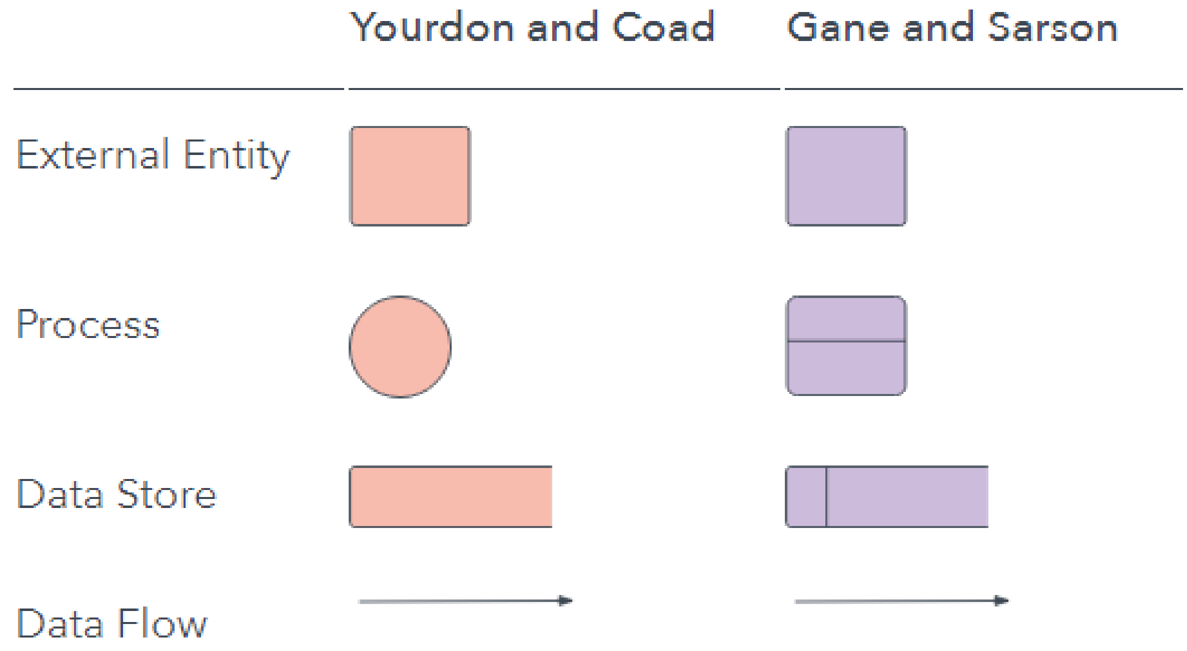
- DFDs are used to show how data flows through a sequence of processing steps.

Data Flow Diagrams

- Elements of a DFD

·There are only four different symbols that are normally used on a DFD.

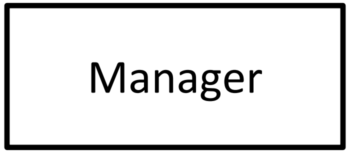
- External entities
- Processes
- Data stores
- Data flows



Data Flow Diagrams - Elements

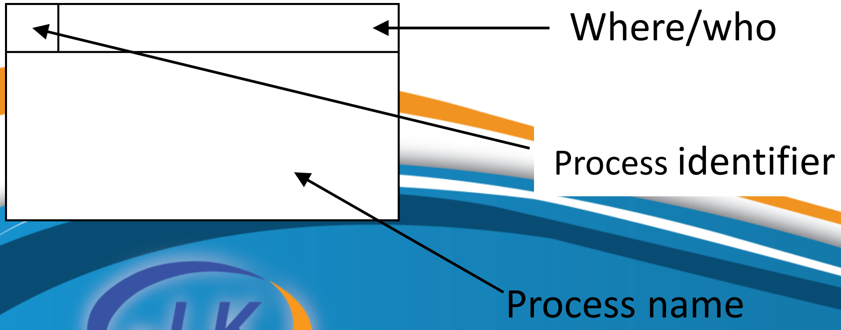
- **External entities**

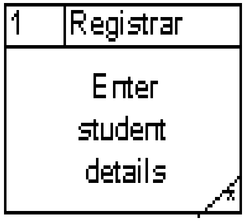
The External Entity symbol represents sources of data to the system or destinations of data from the system.

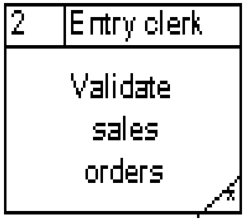


- **Processes**

The Process symbol represents an activity that transforms or manipulates the data (combines, reorders, converts, etc.).



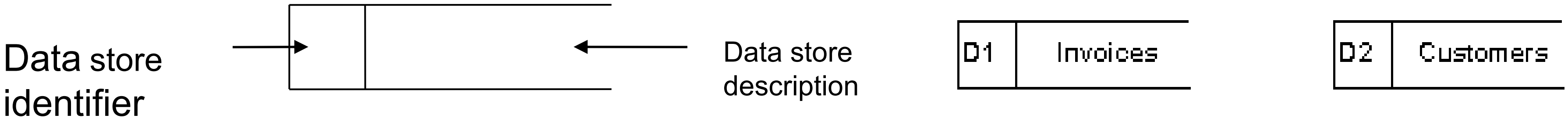




Data Flow Diagrams - Elements

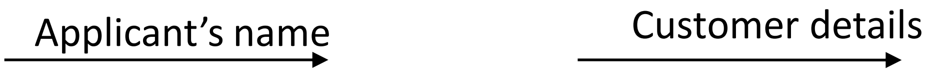
- **Data stores**

The Data Store symbol represents data that is not moving (delayed data at rest).



- **Data flows**

The Data Flow symbol represents movement of data



Data Flow Diagrams - Rules

- Each process must have a minimum of one data flow going into it and one data flow leaving it.

- Each data store must have at least one data flow going into it and one data flow leaving it.

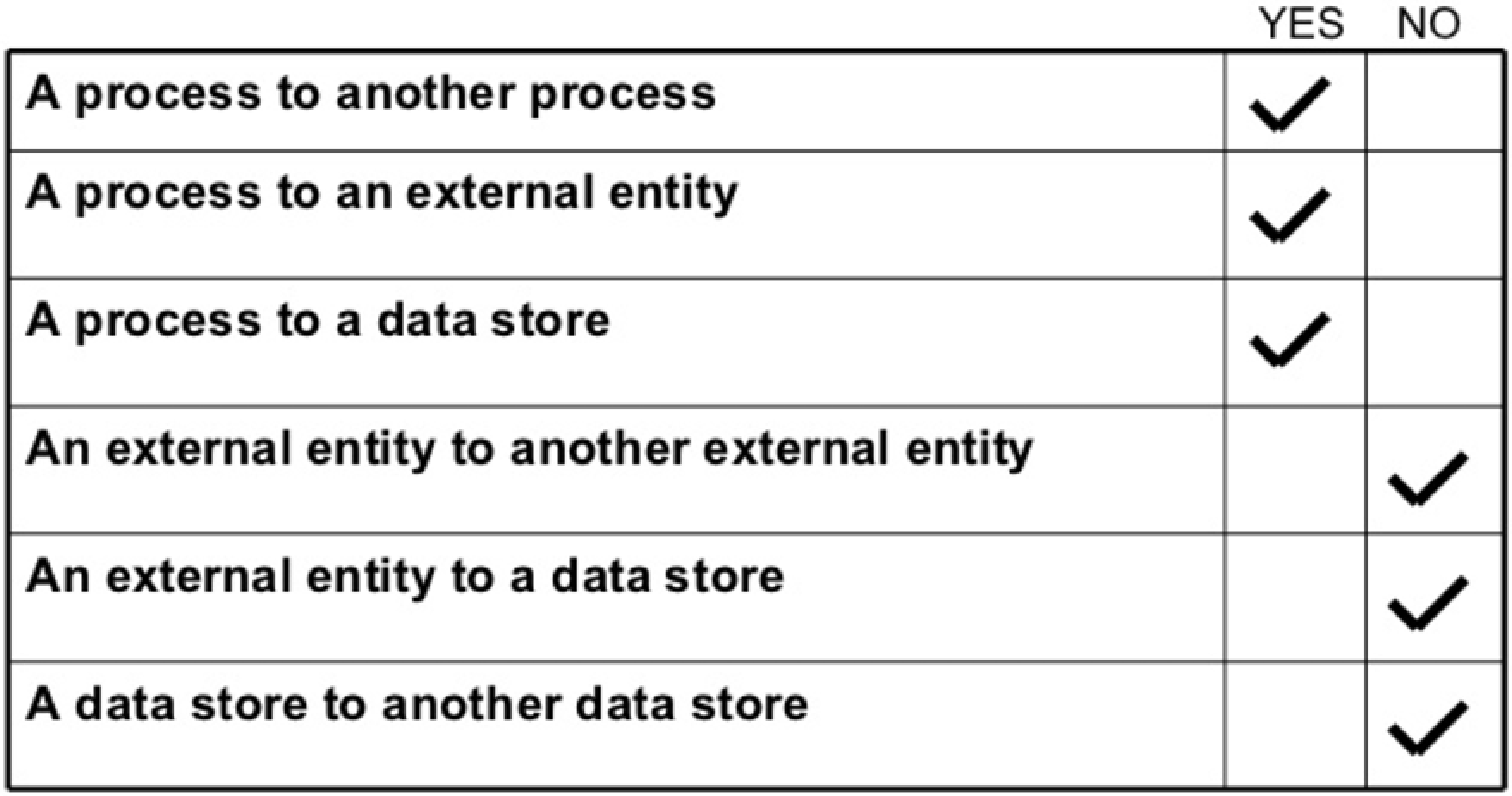
- A data flow out of a process should have some relevance to one or more of the data flows into a process.

Data Flow Diagrams - Rules

- Data stored in a system must go through a process.
- Filing systems within an organization cannot logically communicate with one another unless there is a process involved.
- All processes in DFD must be linked to either another process or a data store.

Data Flow Diagrams - Rules

- Data Flow That Connects



	YES	NO
A process to another process	✓	
A process to an external entity	✓	
A process to a data store	✓	
An external entity to another external entity		✓
An external entity to a data store		✓
A data store to another data store		✓

Context Diagram – 0 Level DFD

What is a Context Diagram

- A DFD that summarizes all processing activity within the system in single process symbol
- The top level view of Information System
- Whole system is represented as one process
- Shows the system boundaries, external entities that interact with the system and major information flows between entities and the system

Context Diagram – (0 Level DFD)

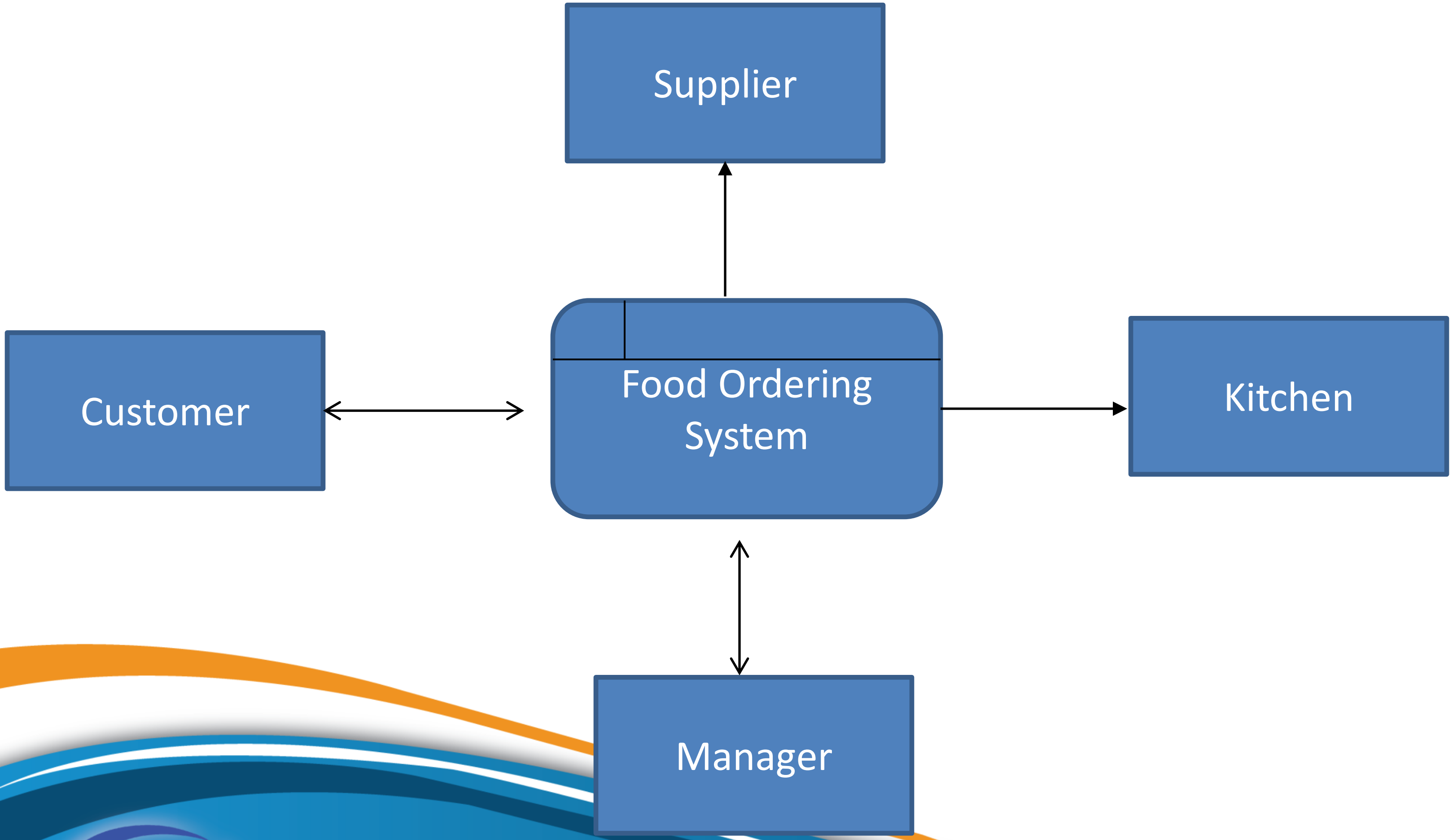
Example – Food Ordering System

Draw a context diagram and level 01 DFD for a food ordering system using below information

- Customer can order foods via the system and system will share the information with kitchen, manager and the supplier to complete the order. Also system can order food, generate reports and order inventory by using the information from the inventory.

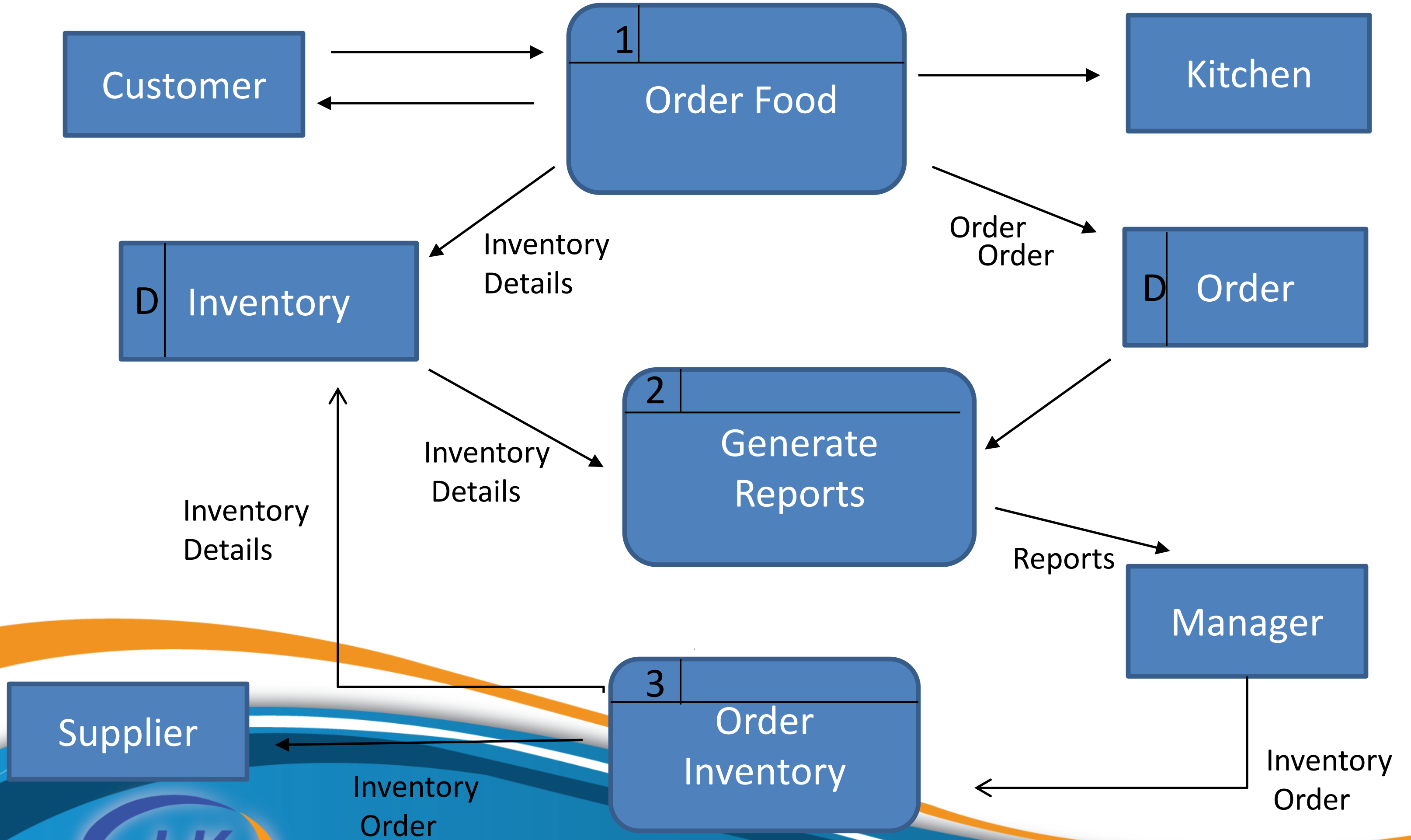
Context Diagram – (0 Level DFD)

Example – Food Ordering System

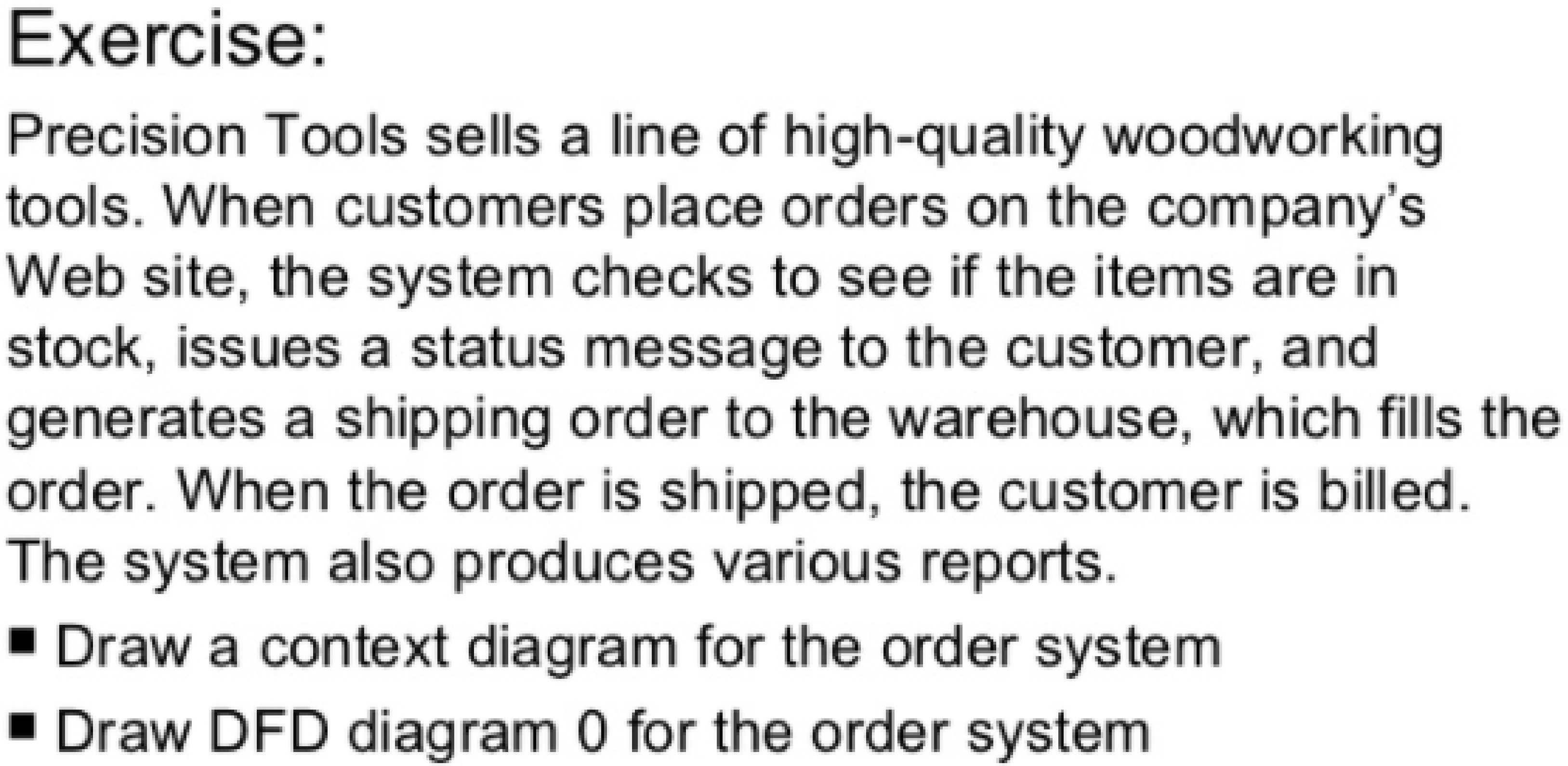


Level 01 DFD

Example – Food Ordering System



Data Flow Diagram

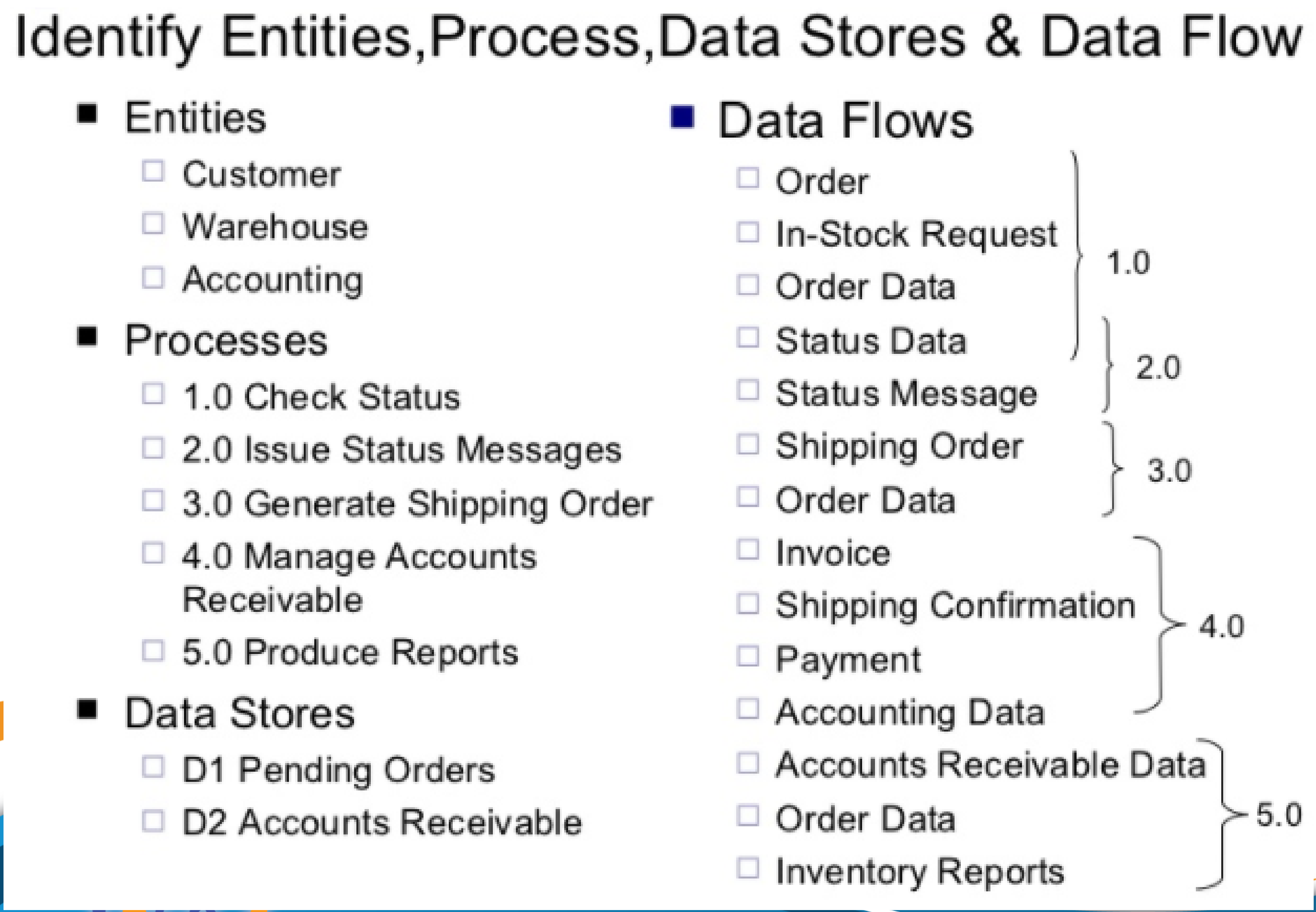


Exercise:

Precision Tools sells a line of high-quality woodworking tools. When customers place orders on the company's Web site, the system checks to see if the items are in stock, issues a status message to the customer, and generates a shipping order to the warehouse, which fills the order. When the order is shipped, the customer is billed. The system also produces various reports.

- Draw a context diagram for the order system
- Draw DFD diagram 0 for the order system

Data Flow Diagram



Identify Entities,Process,Data Stores & Data Flow

- Entities
  - □ Customer
  - □ Warehouse
  - □ Accounting
- Processes
  - □ 1.0 Check Status
  - □ 2.0 Issue Status Messages
  - □ 3.0 Generate Shipping Order
  - □ 4.0 Manage Accounts Receivable
  - □ 5.0 Produce Reports
- Data Stores
  - □ D1 Pending Orders
  - □ D2 Accounts Receivable
- Data Flows
  - □ Order
  - □ In-Stock Request
  - □ Order Data
  - □ Status Data

  1.0

  - □ Status Data
  - □ Status Message

  2.0

  - □ Shipping Order
  - □ Order Data

  3.0

  - □ Invoice
  - □ Shipping Confirmation
  - □ Payment
  - □ Accounting Data

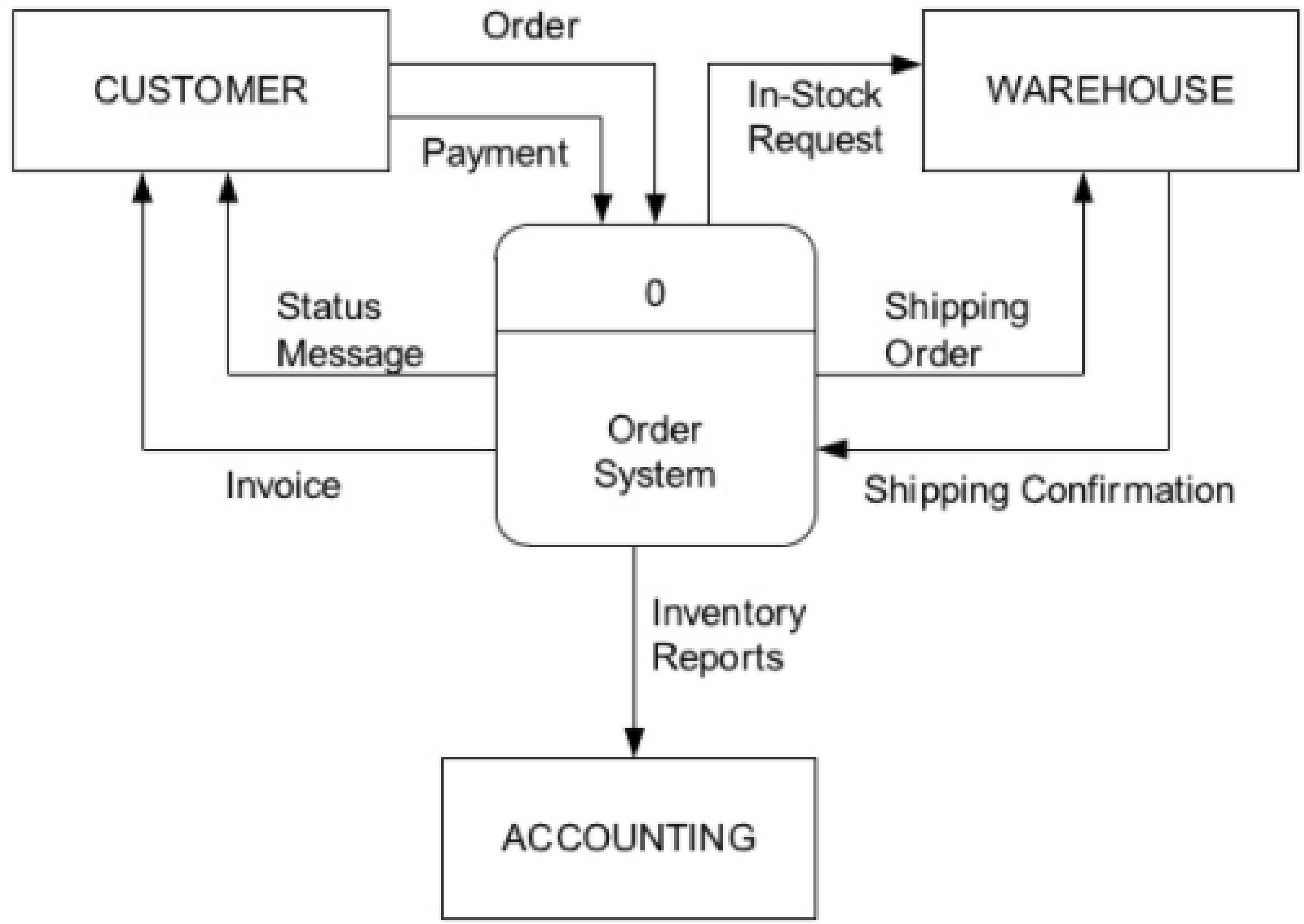
  4.0

  - □ Accounts Receivable Data
  - □ Order Data
  - □ Inventory Reports

  5.0

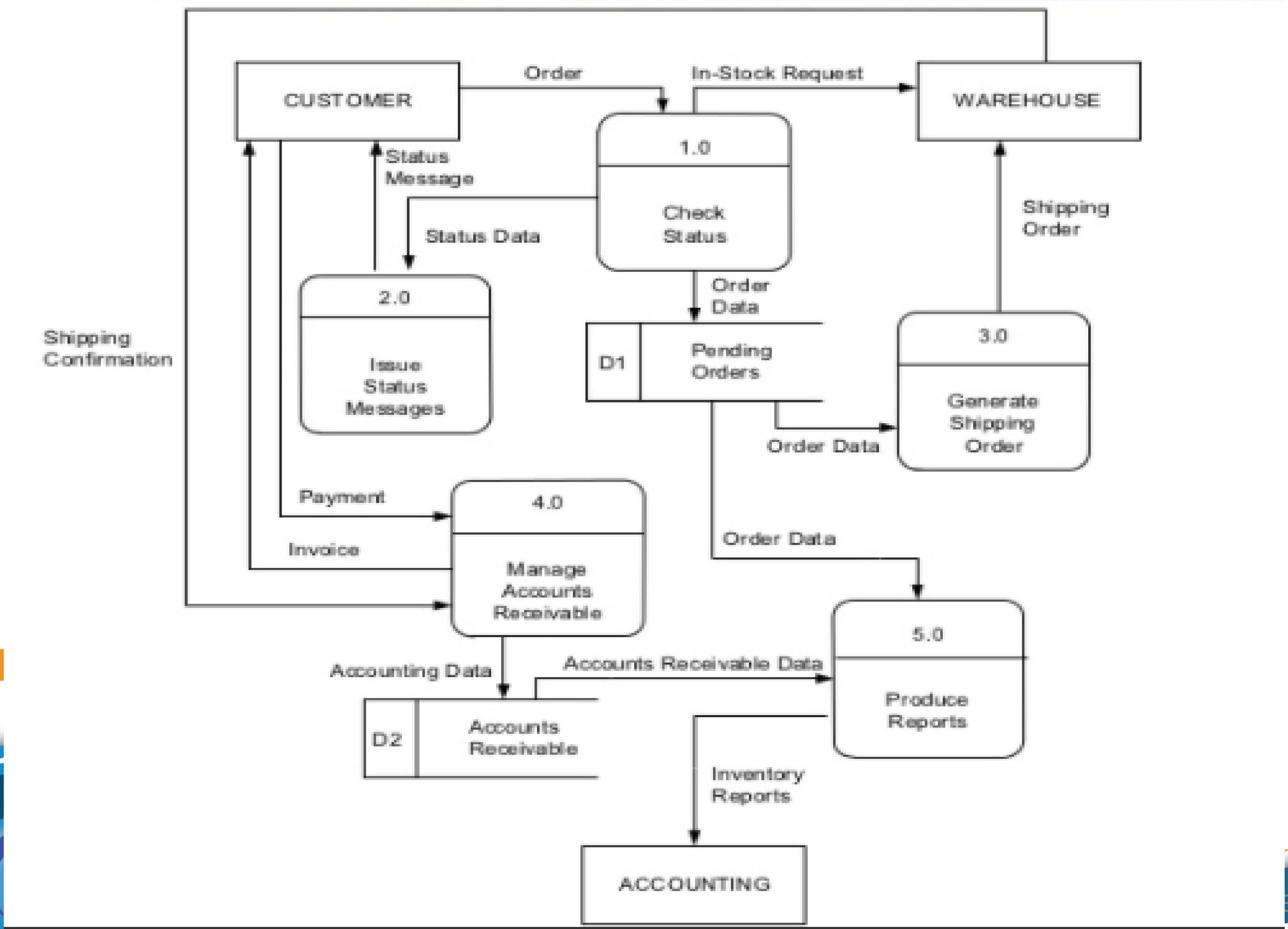
Data Flow Diagram

Context Diagram

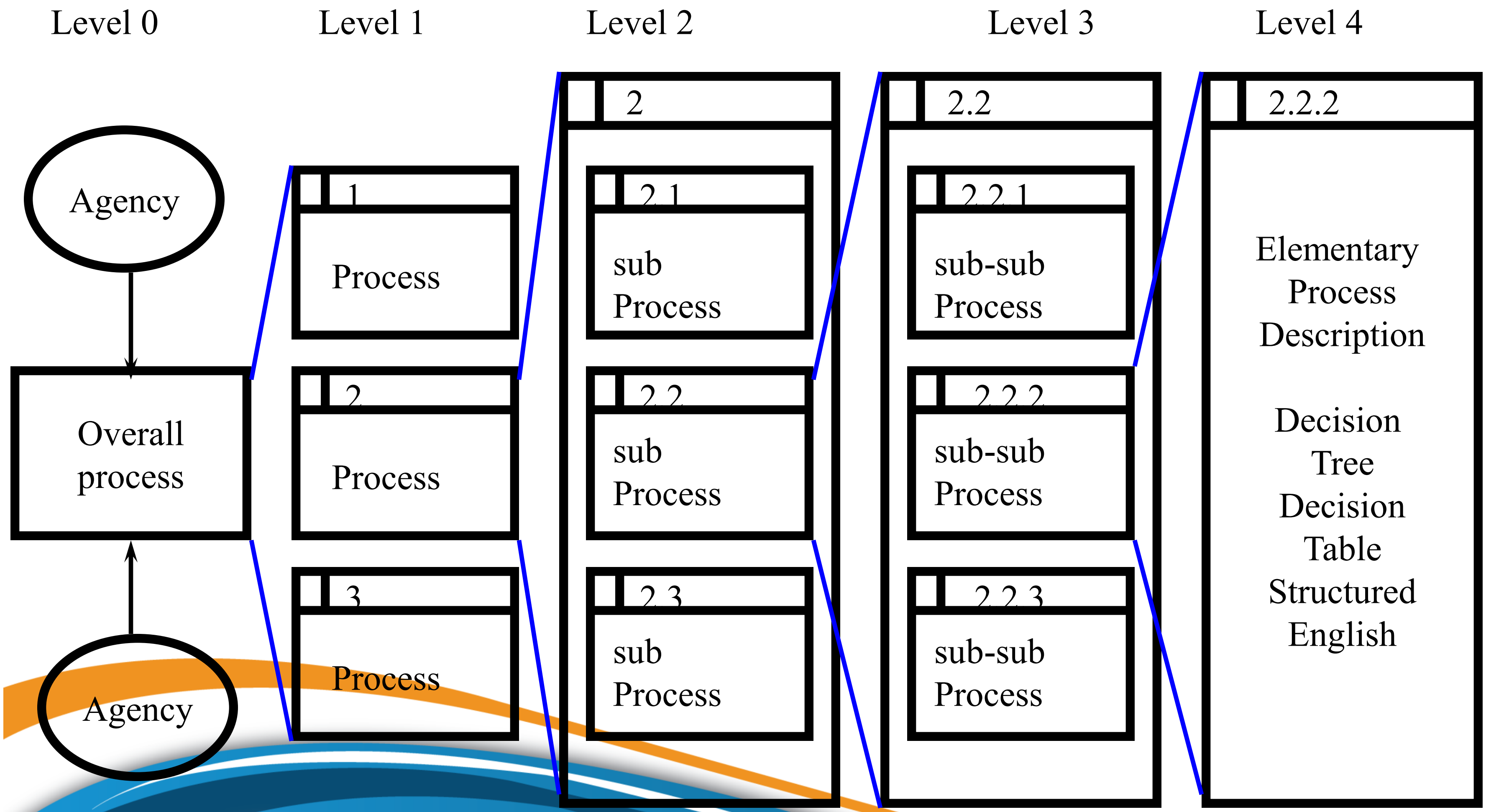


Data Flow Diagram

Level 01 DFD



Components of a Data Flow Model



Level 0

Level 1

Level 2

Level 3

Level 4

Agency

Overall process

Agency

1 Process

2 Process

3 Process

2

2.1 sub Process

2.2 sub Process

2.3 sub Process

2.2

2.2.1 sub-sub Process

2.2.2 sub-sub Process

2.2.3 sub-sub Process

2.2.2

Elementary Process Description

Decision Tree

Decision Table

Structured English

Summery

In this section you have given an idea about system analysis and system design activities.

Also this section covered the requirement analysis activities.

References

- SOFTWARE ENGINEERING, Ninth Edition, Ian Sommerville

Next Lesson >>>

Lesson 05 : System Design – Part 02