

LAB I. DATA FLOW DIAGRAM. (DFD)

01418321 System Analysis and Design
Chalothon Chootong (Ph.D.)

Department of Computer Science and Information, Faculty of Science
at Sriracha, Kasetsart University Sriracha Campus

chootong.c@ku.th

Data flow diagram (DFD)

- Data flow diagrams show how data is processed by a system in terms of inputs and outputs. Components of data flow diagram includes
 - Process
 - Flow
 - Store
 - Terminator
- A logical data flow diagram shows system's activities while a physical data flow diagram shows a system's infrastructure
- A data flow diagram can be designed early in the requirement elicitation process of the analysis phase within the **SDLC (System Development Life Cycle)** to define the project scope.

What is a data flow diagram?

- It uses defined symbols like **rectangles**, **circles** and **arrows**, plus **short text** labels, to show **data inputs**, **outputs**, **storage points** and the **routes** between each destination.
- Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.
- They can be used to analyse an **existing system** or model a new one.

Developer to CEO

History of the DFD

- Data flow diagrams were popularized in the late 1970s, arising from the book *Structured Design*.
- The structured design concept took off in the software engineering field, and the DFD method took off with it.
- It became more popular in business circles, as it was applied to business analysis, than in academic circles.

History of the DFD

- Also contributing were two related concepts:
 - **Object Oriented Analysis and Design (OOAD)**, put forth by Yourdon and Peter Coad to analyze and design an application or system.
 - **Structured Systems Analysis and Design Method (SSADM)**, a waterfall method to analyze and design information systems.

Three other experts contributing to this rise in DFD methodology were **Tom DeMarco**, **Chris Gane** and **Trish Sarson**. They teamed up in different combinations to be the main definers of the symbols and notations used for a data flow diagram.

Symbols and Notations Used in DFDs

■ Yourdon and Coad

■ Gane and Sarson

Notation

Yourdon and Coad

Gane and Sarson

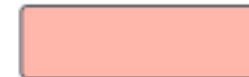
External Entity



Process



Data Store



Data Flow



Symbols and Notations Used in DFDs

- **External entity:**

- An outside system that **sends** or **receives data**, **communicating** with the system being diagrammed.
- They are the **sources** and **destinations** of information entering or **leaving** the system.
- They might be an **outside organization** or **person**, a **computer system** or a **business system**.
- terminators, sources or actors

Symbols and Notations Used in DFDs

- **Process:**

- Any process that changes the data, producing an output.
- It might perform computations, or sort data based on logic, or direct the data flow based on business rules.
- A short label is used to describe the process, such as “Submit payment.”

- **Data store:**

- Files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as “Orders.”

Symbols and Notations Used in DFDs

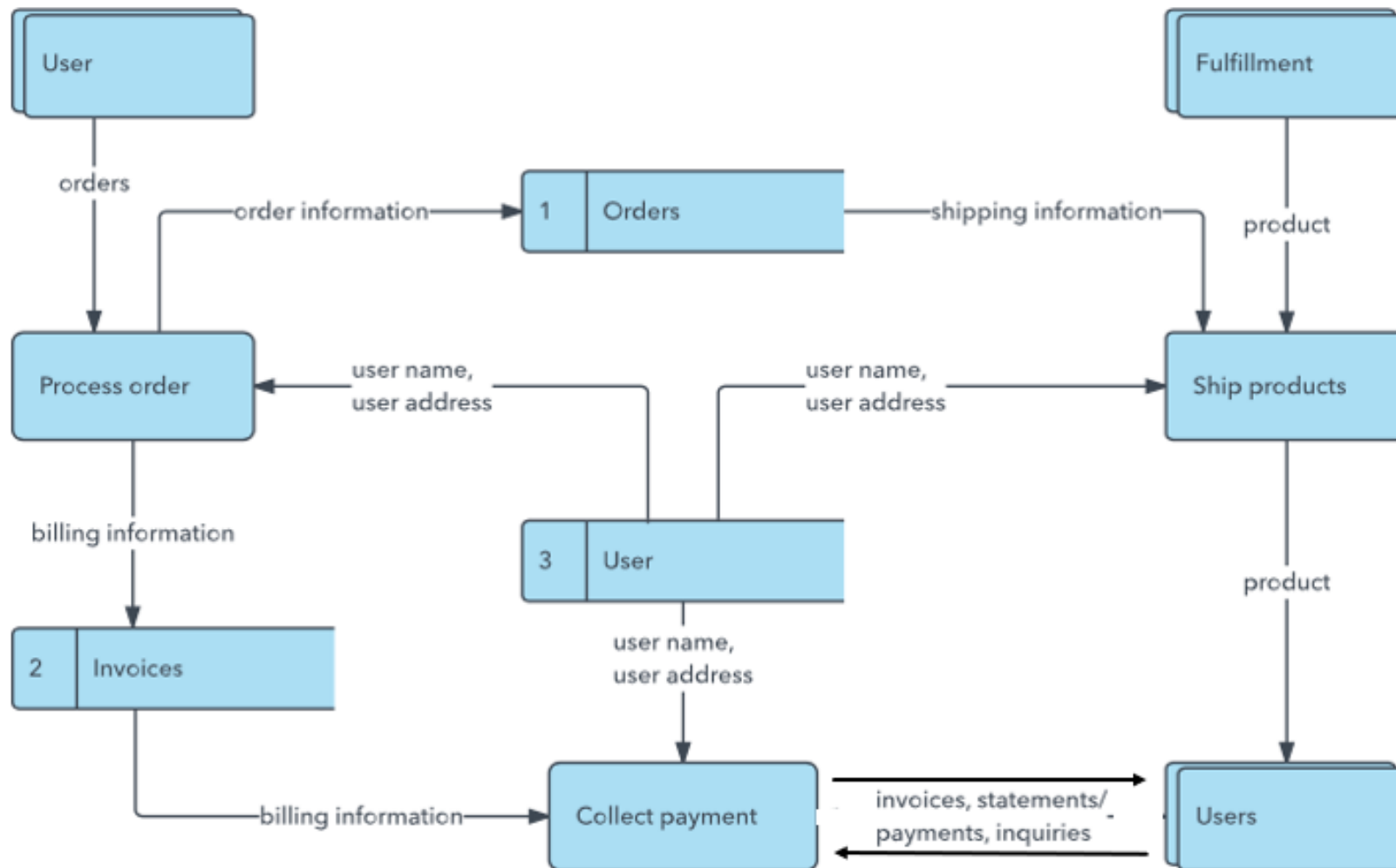
- **Data flow:**

- The route that data takes between the external entities, processes and data stores.
- It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like “Billing details.”

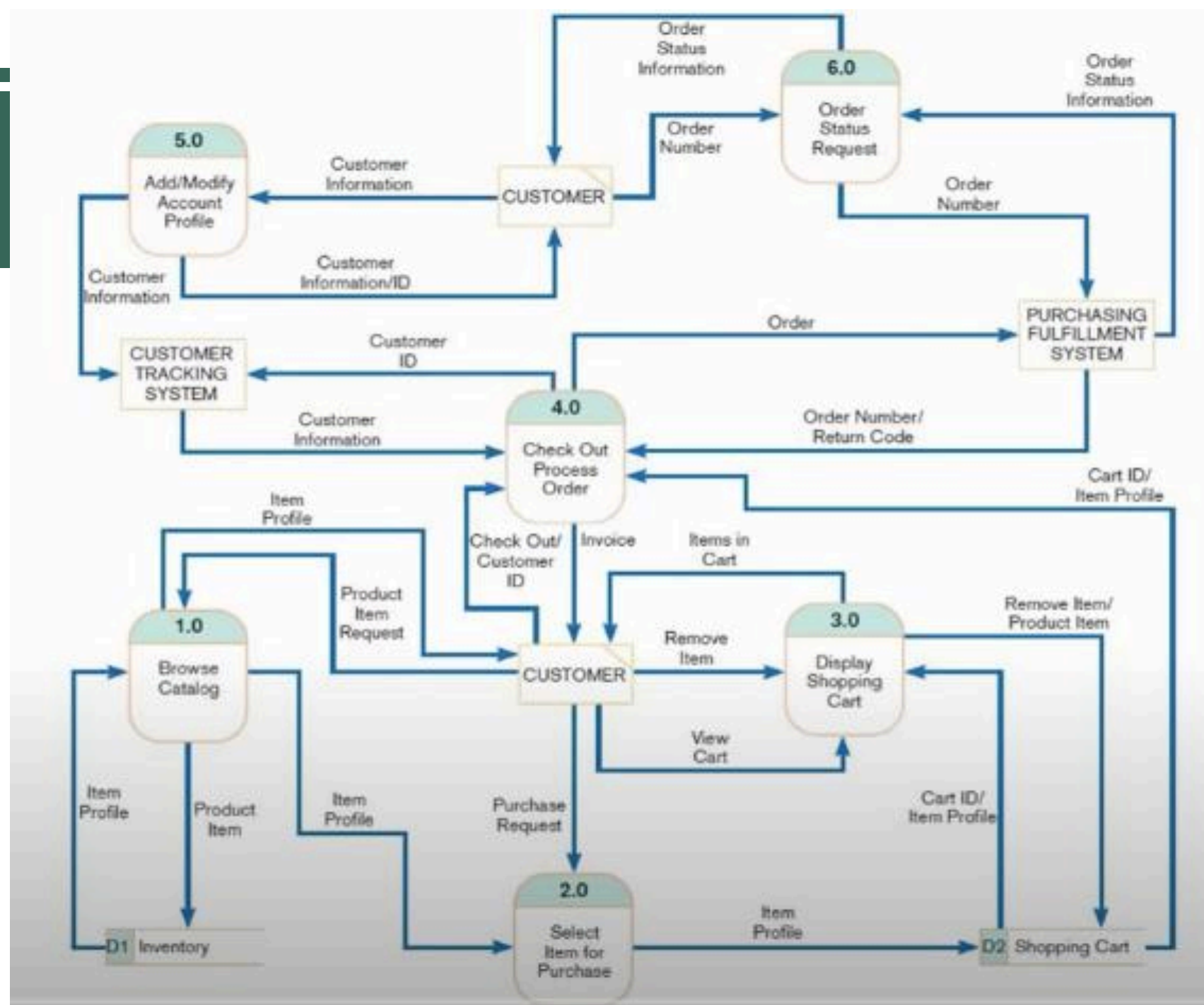
- **DFD rules and tips**

- Each process should have at least one input and an output.
- Each data store should have at least one data flow in and one data flow out.
- Data stored in a system must go through a process.
- All processes in a DFD go to another process or a data store.

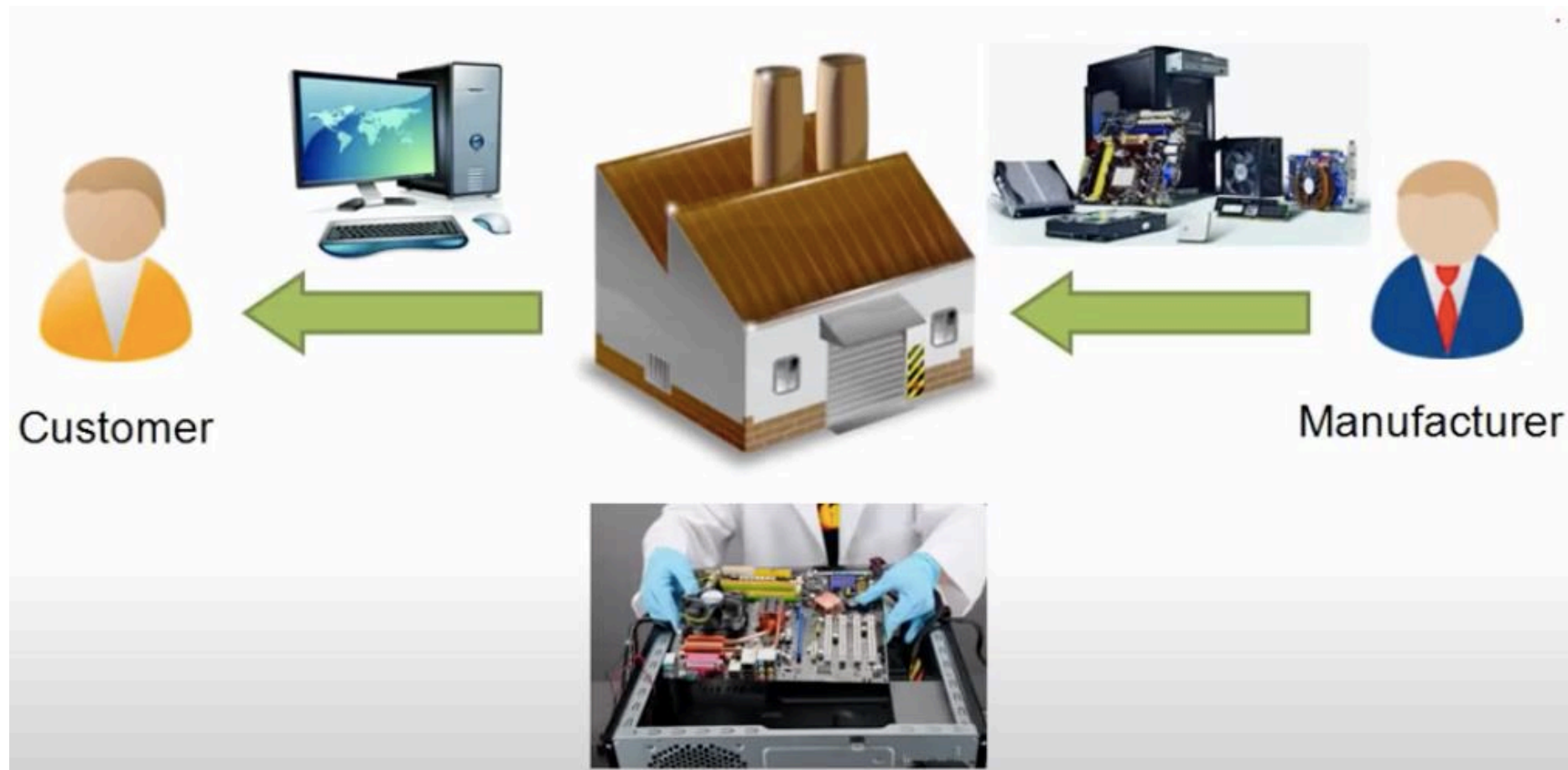
DFD Example



Reading Data Flow Diagrams



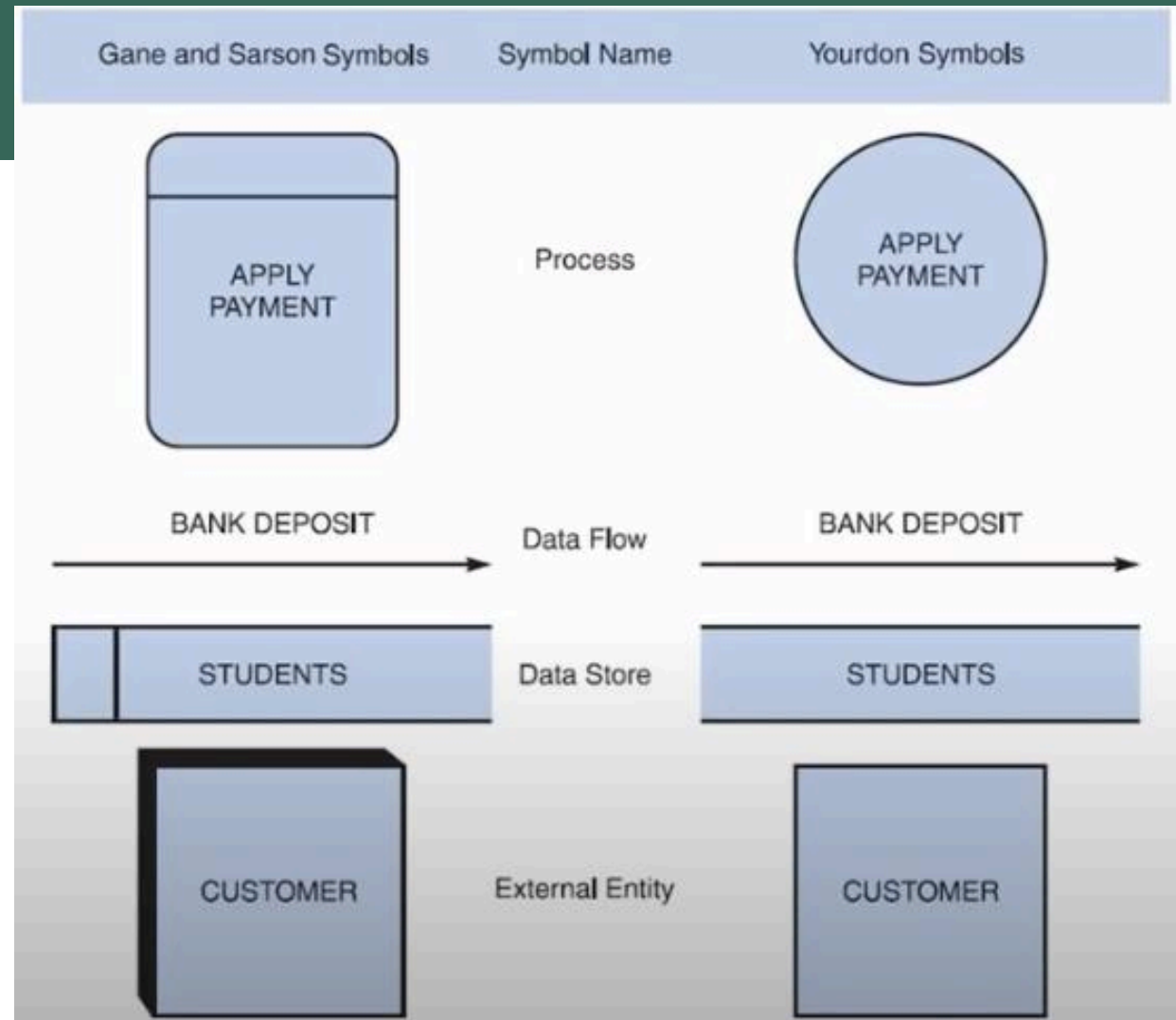
Factory



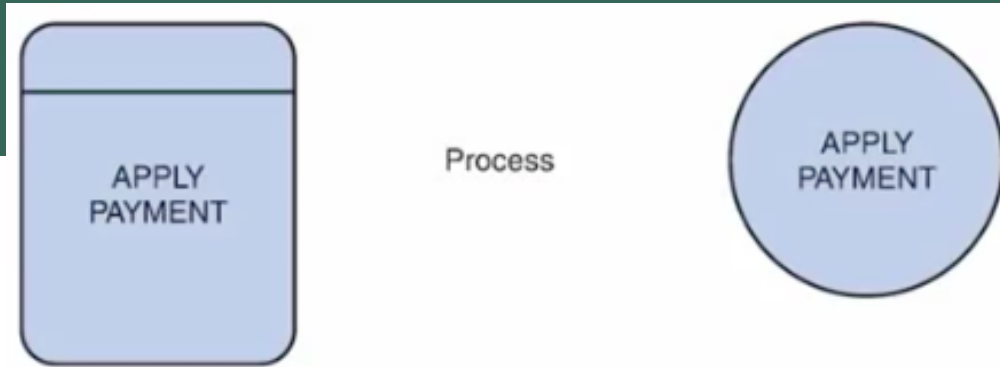
ข้อดีของ Data Flow Diagram

1. แผนภาพกระแสข้อมูล ใช้งานได้อิสระโดยไม่จำเป็นต้องใช้เทคนิคอื่น ๆ เข้ามาช่วย
2. การใช้แผนภาพกระแสข้อมูล ใช้งานได้ง่าย
3. เป็นเครื่องมือในการสื่อสารระหว่างทีมงานที่พัฒนาระบบ และผู้ใช้งานระบบได้เป็นอย่างดี
4. แผนภาพกระแสข้อมูล ทำให้ขั้นตอนการทำงาน และข้อมูลต่าง ๆ เป็นแผนภาพการไหลของข้อมูลระหว่างโปรเซสได้

Data Flow Diagram Symbols



การวิเคราะห์และเขียน Process



- ขั้นตอนหรือกระบวนการทำงานในการรับข้อมูลและจัดทำผลลัพธ์
 - ใน DFDs จะไม่แสดงรายละเอียดการทำงานภายในของ process การแสดงรายละเอียดการทำงานของแต่ละ process จะอธิบายในส่วนของ process descriptions
 - ผลลัพธ์ที่ออกมาจาก process จะต้องมีความแตกต่างของข้อมูลที่ได้รับเข้ามาใน process เดียวกัน

Data Flow

- เส้นทางการไหลของข้อมูล (Data Flows) เป็นการสื่อสารระหว่างขั้นตอนการทำงาน (Process) ต่างๆ และสภาพแวดล้อมภายนอกหรือภายในระบบ
- โดยแสดงถึงข้อมูลที่นำเข้าไปในแต่ละ Process และข้อมูลที่ส่งออกจาก Process
- ใช้ในการแสดงถึงการบันทึกข้อมูล การลบข้อมูล การแก้ไขข้อมูลต่างๆ ในไฟล์หรือในฐานข้อมูล "Data Store"

Data Store



- แหล่งจัดเก็บข้อมูล (Data Store) เป็นแหล่งเก็บ/บันทึกข้อมูล
เปรียบเสมือนคลังข้อมูล (เทียบเท่ากับไฟล์ข้อมูล และฐานข้อมูล)
โดยอธิบายรายละเอียดและคุณสมบัติเฉพาะตัวของสิ่งที่ต้องการเก็บ/
บันทึก

External Entity

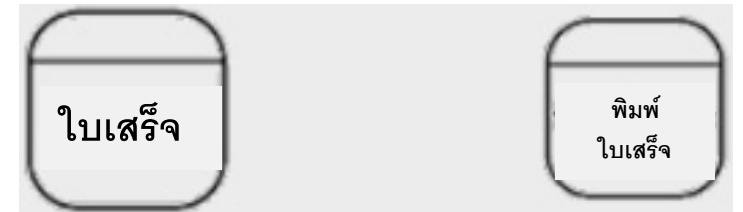
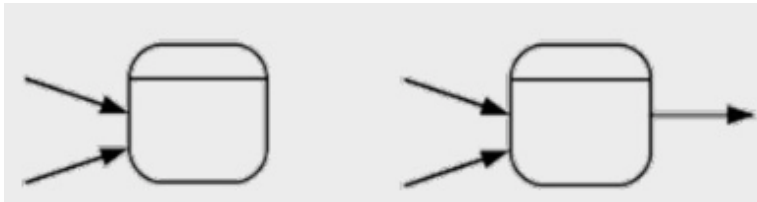


- หมายถึง บุคคล หน่วยงานในองค์กร องค์กรอื่นๆ หรือระบบงานอื่นๆ ที่อยู่ภายนอกขอบเขตของระบบ แต่มีความสัมพันธ์กับระบบ โดยมีการส่งข้อมูลเข้าสู่ระบบเพื่อดำเนินงาน และรับข้อมูลที่ผ่านการดำเนินงานเรียบร้อยแล้วจากระบบ
- หรือเรียกว่า External Agent หรือ Source/Sink

DFD Rules

■ Process

- A. No process can have only outputs
- B. No process can have only inputs
- C. A process has a verb and phrase label



D1

พิมพ์ใบเสร็จ

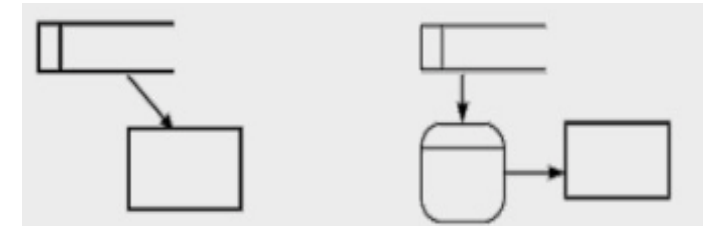
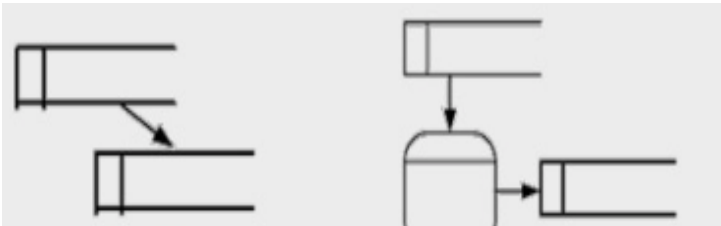
D1

แฟ้มใบเสร็จ

DFD Rules

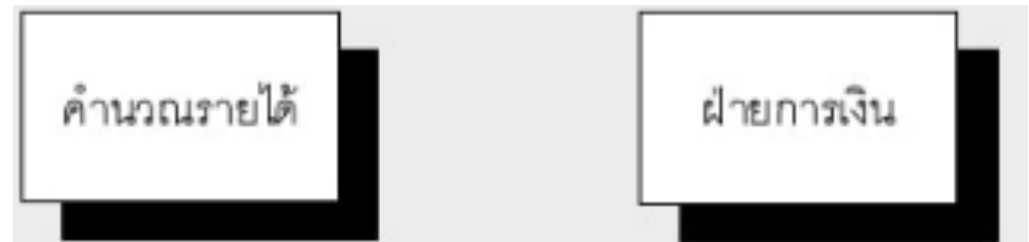
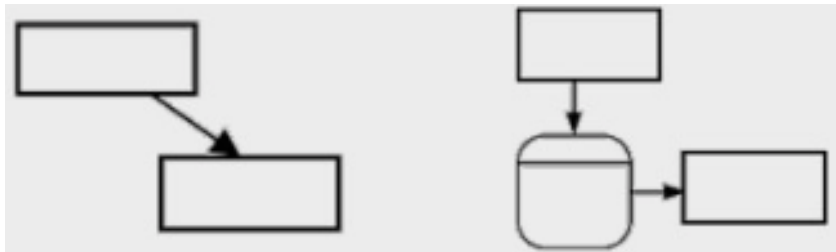
■ Data Store

- D. Data cannot be moved from one store to another
- E. Data cannot move from an outside source to a data store
- F. Data cannot move directly from a data store to a data sink
- G. Data store has a noun phrase label



DFD Rules

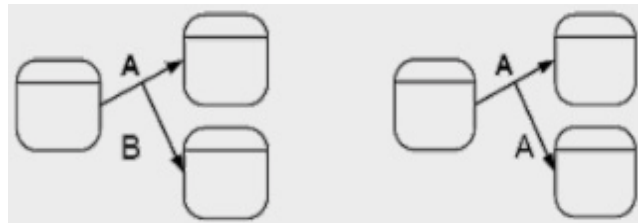
- Source/Sink
 - **H.** Data cannot move directly from a source to a sink
 - **I.** A source/sink has a noun phrase label



DFD Rules

■ Data Flow

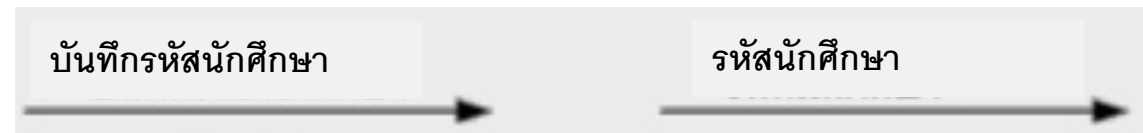
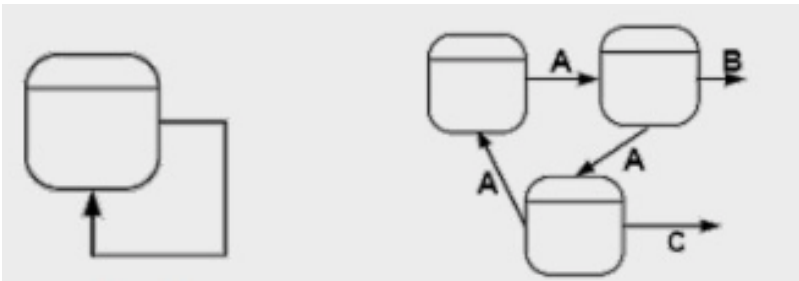
- **J.** A data flow has only one direction of flow between symbols
- **K.** A fork means that exactly the same data go from a common location to two or more processes, data stores, or source/sinks
- **L.** A joint means that exactly the same data come from any two or more different processes, data store or source/sinks to common location.



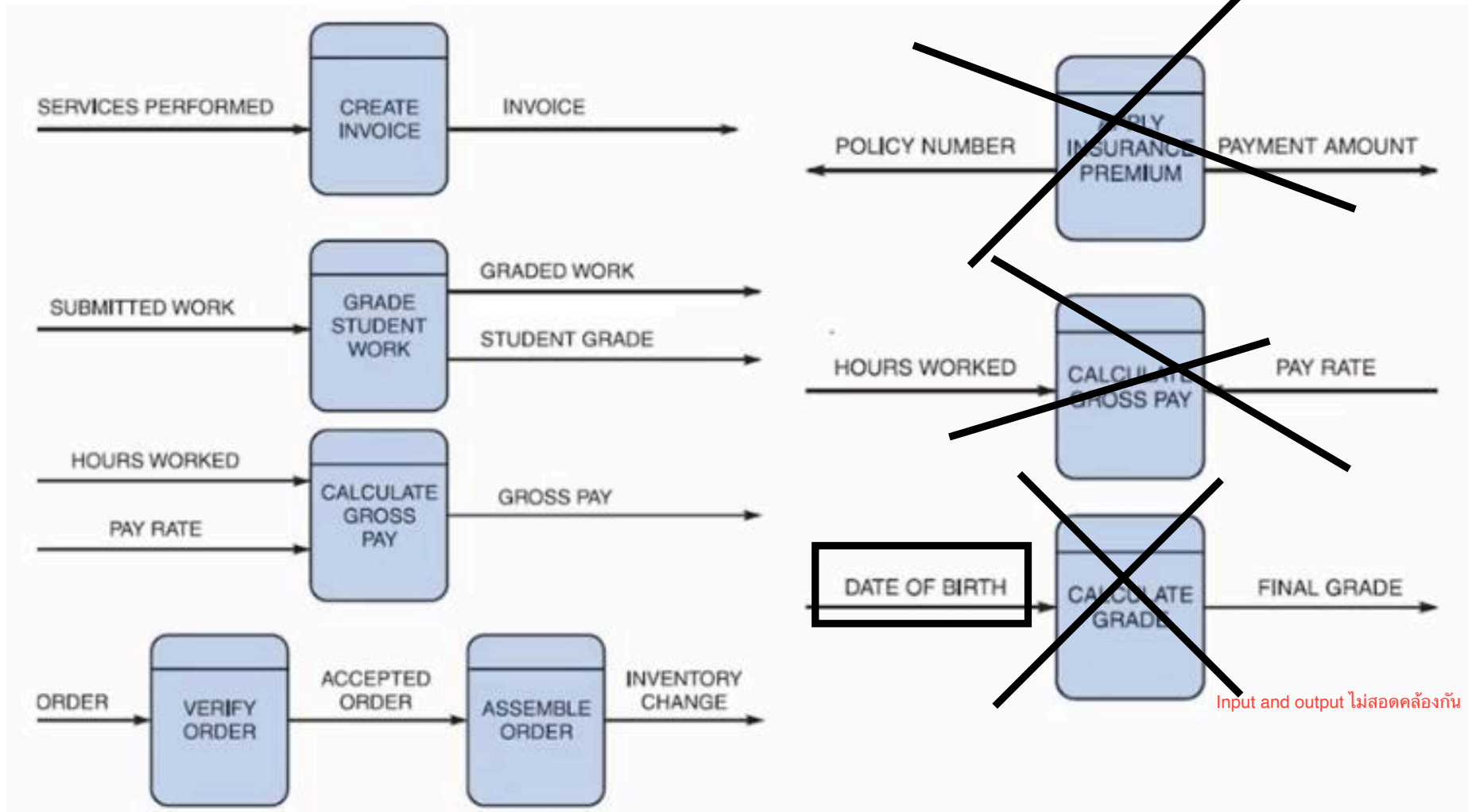
DFD Rules

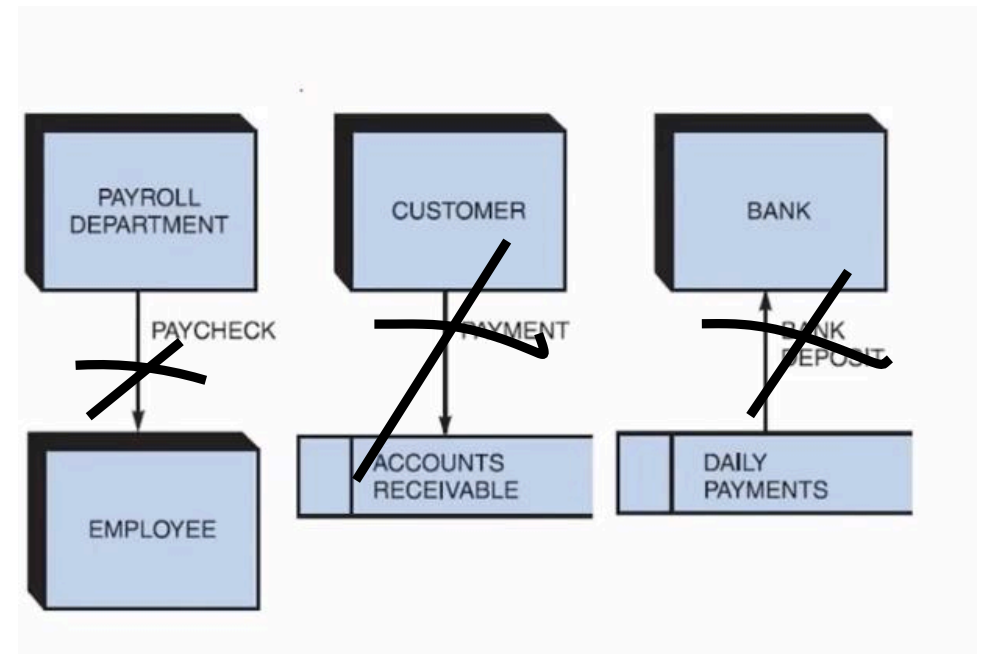
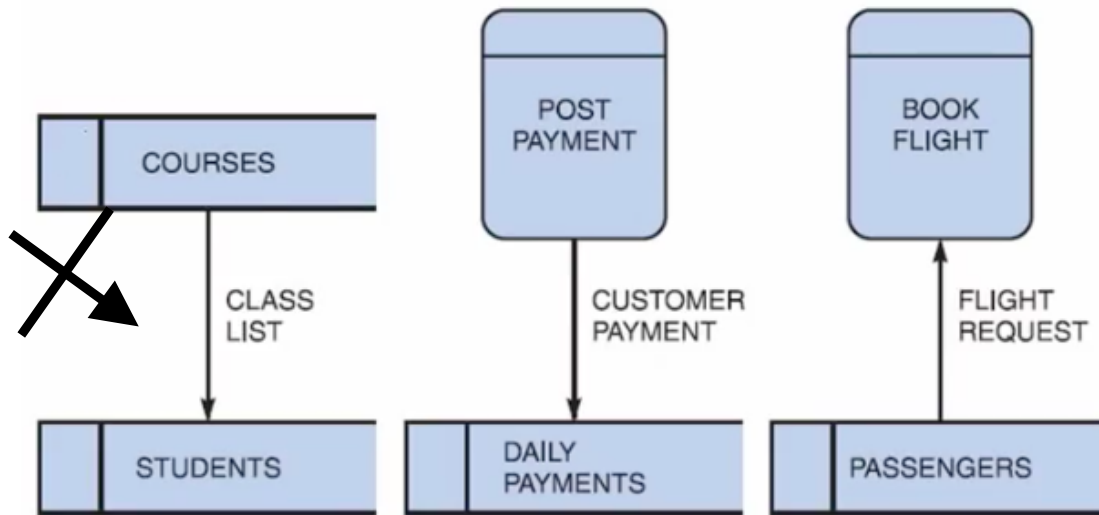
■ Data Flow

- **M.** A data flow cannot go directly back to the same process it leaves.
- **N.** A data flow to a data store means update
- **O.** A data flow from a data store means retrieve or use.
- **P.** A data flow has a noun phrase label



DFD Example





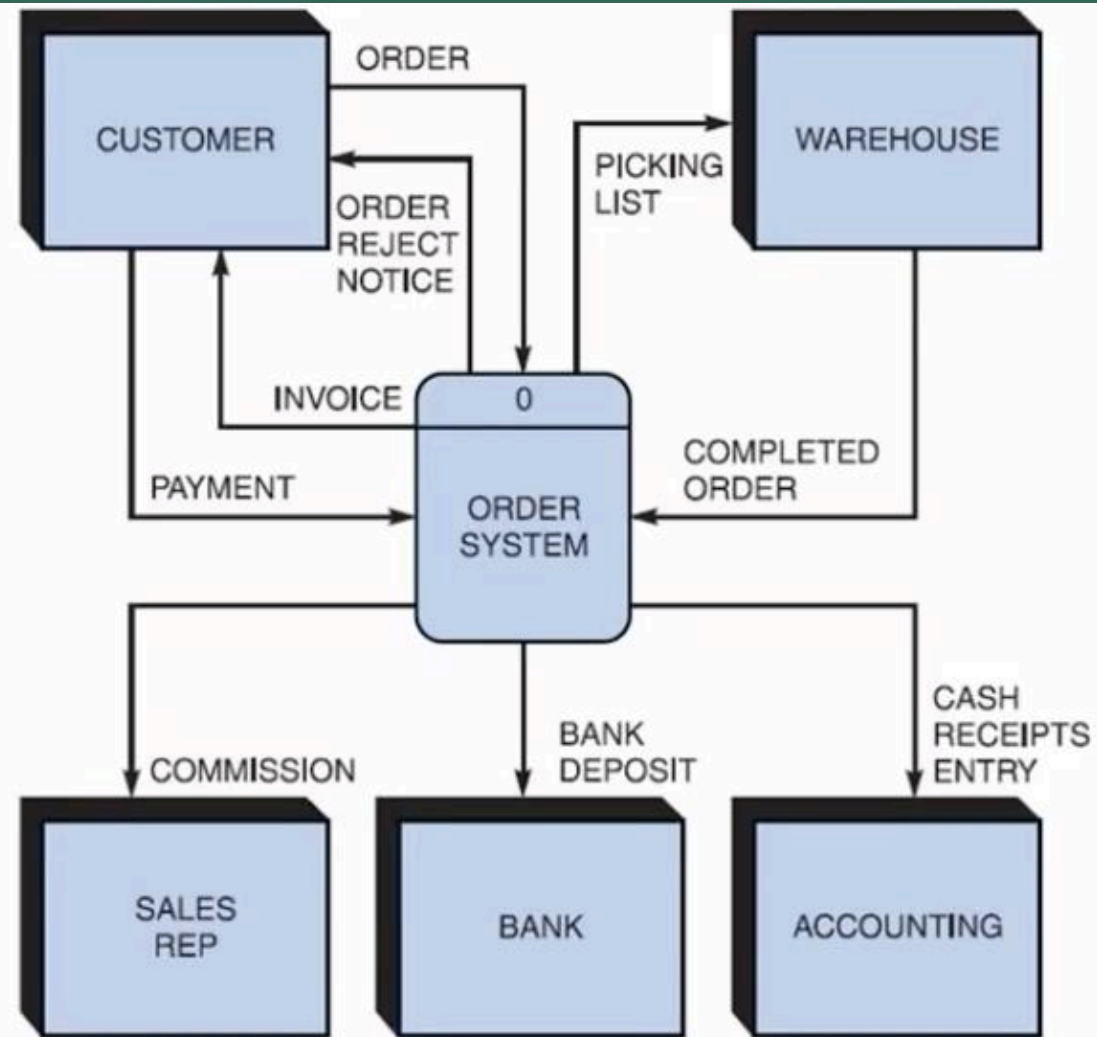
Context Diagram

No storage

คือ ภาพระดับสูงสุด (Top Level) **ศึกษาถึงลักษณะโดยรวมของระบบ** ภาพแสดงขอบวนการหลักเพียง 1 ขบวนการ ขบวนการดังกล่าวถือเป็นหัวใจของการทำงานเพราะแทนการทำงานทั้งระบบ

- แผนภาพระดับแรกของ DFDs แสดงภาพรวมของระบบ โดยแสดงหน่วยภายนอกที่เกี่ยวข้อง
- จะประกอบด้วย 1 process เท่านั้น และ process ดังกล่าวนี้นี้จะมีชื่อเป็นชื่อของระบบและมีหมายเลขประจำ process เป็นหมายเลข 0
- context diagram จะมีเพียงสามสัญลักษณ์ คือ external, process (1 process) และ data flow (จะไม่เขียน data store ในระดับนี้)

ตัวอย่าง Context Diagram

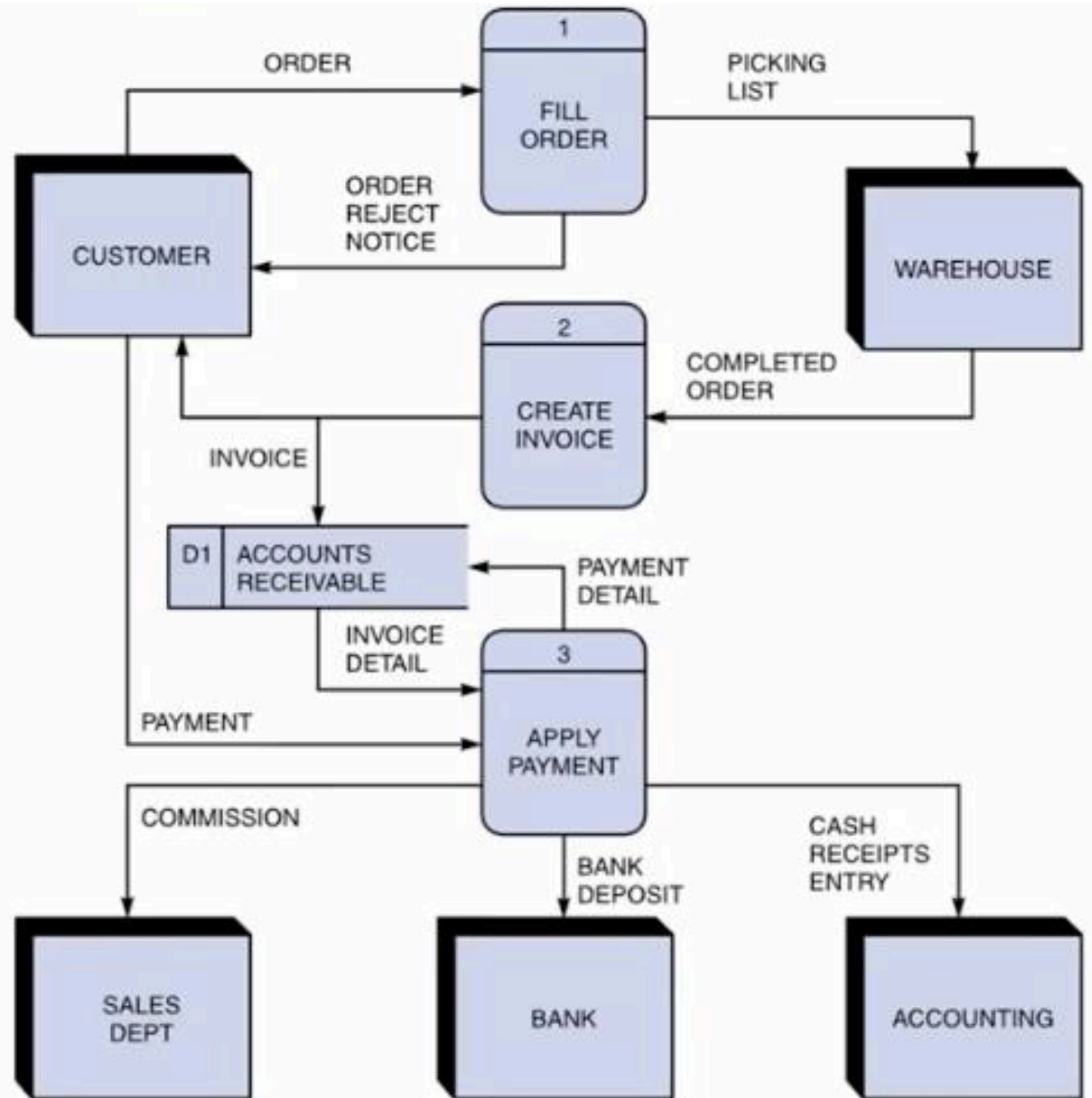


ข้อแนะนำในการเขียน Context Diagram

- เขียน Context Diagram ภายใน 1 หน้า
- ชื่อ Process ใน Context Diagram จะเป็นชื่อระบบงาน
- ห้ามใช้ชื่อซ้ำในการอ้างถึงแต่ละสัญลักษณ์
- อย่าเขียนเส้นข้ามหรือทับกัน
- ใช้หมายเลขอ้างอิงเดียวกันในแต่ละสัญลักษณ์ของ Process

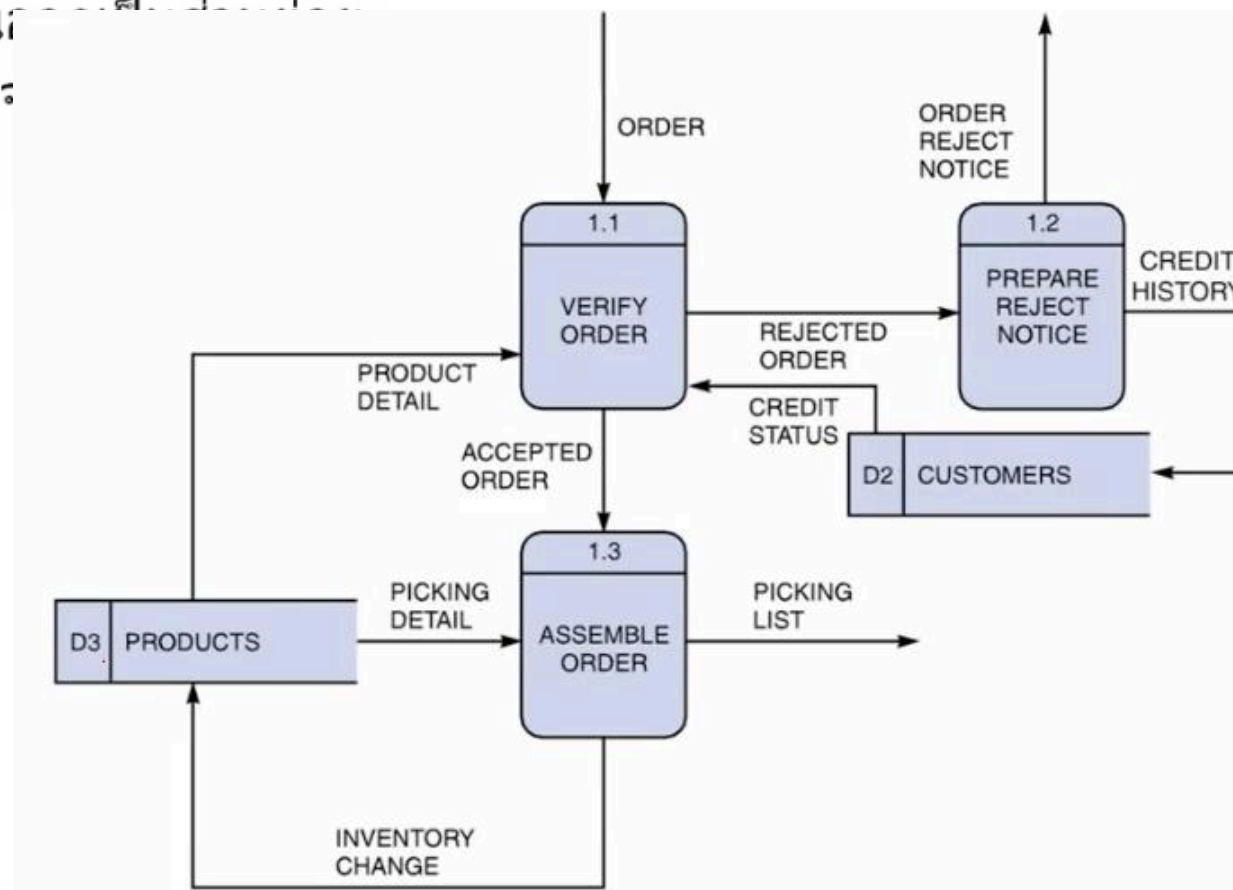
- ~~Level-1~~ Diagram คือ แผนภาพกระแสข้อมูลในระดับที่แสดงขั้นตอนการทำงานหลักทั้งหมด (Process หลัก) ของระบบแสดงทิศทางการไหลของ Data Flow และแสดงรายละเอียดของแหล่งจัดเก็บข้อมูล (Data Store)
- Level-0 Diagram เป็นการแสดงให้เห็นถึงรายละเอียดของ Process การทำงานหลักๆ ที่มีอยู่ภายในภาพรวมของระบบ (Context Diagram) ว่ามีขั้นตอนใดบ้าง โดยแต่ละ Process จะมีหมายเลขกำกับอยู่ด้านบนของสัญลักษณ์ ตั้งแต่ 1 เป็นต้นไป
- อาจแตกย่อยได้ 3-9 ขบวนการ

Level I



Decomposition of DFD

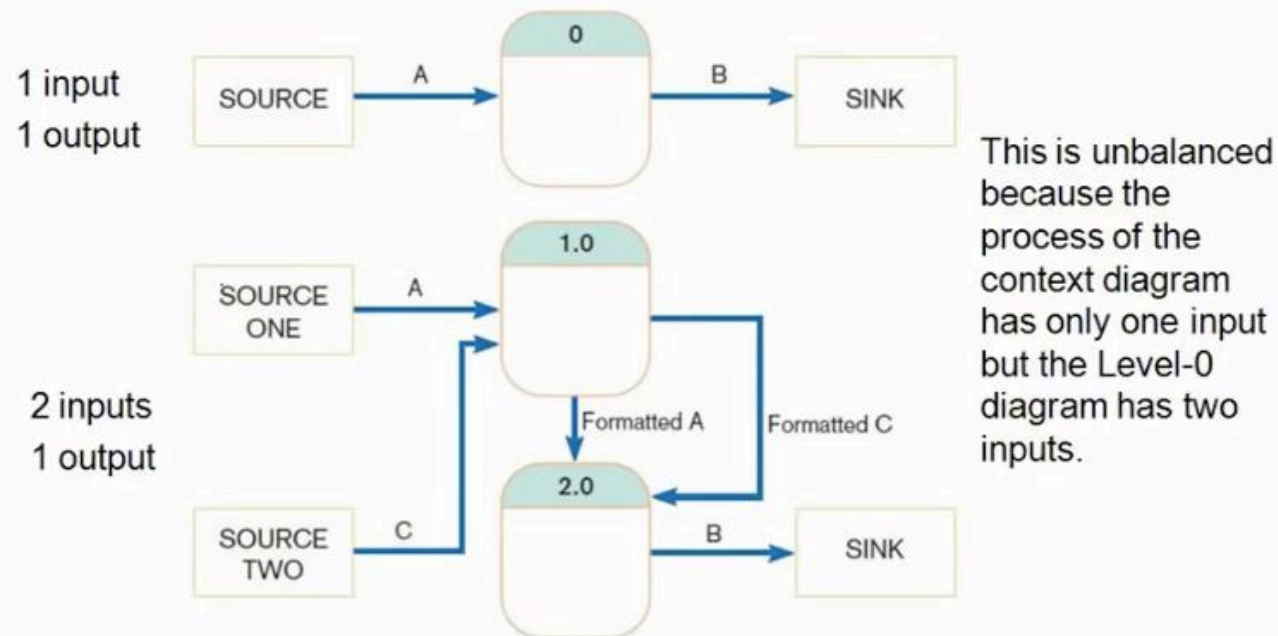
คือ การแบ่ง/แยก/ย่อยระบบและขั้นตอนการทำงาน
โดยในแต่ละขั้นตอนที่แยกออกมา (Subsystems) จะ
รายละเอียดของการทำงานเพิ่มมากขึ้น



Balancing DFD

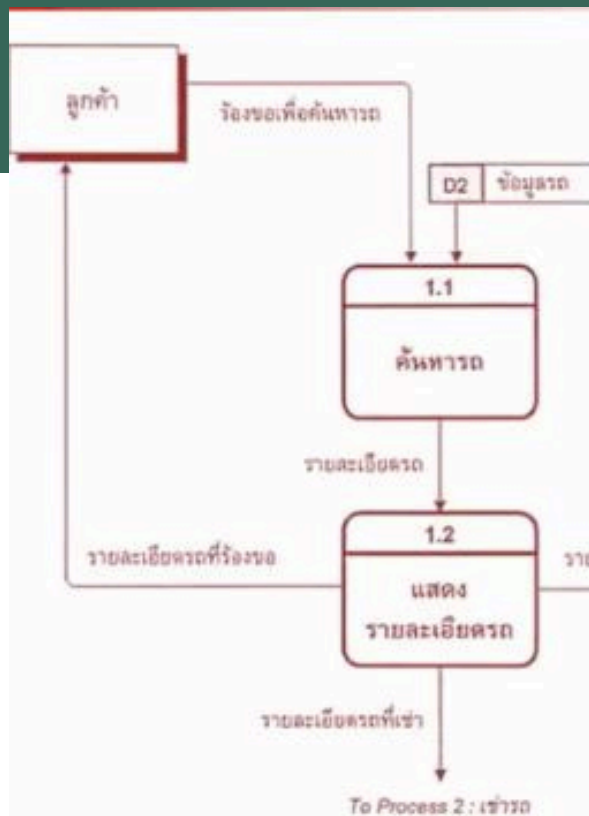
- เป็นคุณลักษณะที่ใช้ในการตรวจสอบความถูกต้องระหว่างกลุ่มของ DFD ทั้งหมด
- โดยใช้การตรวจสอบจาก input data flow ของ Parent และ output data flows เปรียบเทียบกับที่ปรากฏใน child (ถ้าใน parent มี ใน child ก็ต้องมีเหมือนกันหรืออ้างอิงกันได้)

An unbalanced set of data flow diagrams



Process Description

- แผนภาพที่สามารถนำเสนอให้เห็นภาพรวมของระบบ
- จุดประสงค์เพื่อ
 - เพื่อลดความไม่ชัดเจนหรือความกำกวมของโปรเซส และทำให้ทราบว่าแต่ละโปรเซสมีขั้นตอนการทำงานอย่างไร
 - เพื่อความเที่ยงตรงและเข้าใจตรงกัน โปรแกรมเมอร์สามารถนำไปใช้เพื่อออกแบบและพัฒนาโปรแกรมได้อย่างถูกต้อง ตรงตามวัตถุประสงค์
 - เพื่อใช้สำหรับการตรวจสอบในขั้นตอนของการออกแบบระบบถึงกระบวนการที่รับข้อมูลนำเข้ามาคืออะไรและประมวลผลออกมาเป็นผลลัพธ์



Process Description

System	ระบบศูนย์บริการรถเช่า
DFD number	1
Process name	ค้นหาและแสดงรายละเอียด
Input data flows	ร้องขอเพื่อค้นหา
Output data flows	รายละเอียดที่ร้องขอ รายละเอียดที่เช่า รายละเอียดที่จอง
Data stores used	ข้อมูล
Description	โปรแกรมหลักที่นำมาใช้เพื่อการค้นหาและแสดงรายละเอียด
Method	1.1 ค้นหา 1.2 แสดงรายละเอียด

Process Description

System	ระบบศูนย์บริการรถเช่า
DFD number	1.1
Process name	ค้นหา
Input data flows	ร้องขอเพื่อค้นหา
Output data flows	รายละเอียด
Data stores used	ข้อมูล
Description	ค้นหาที่ต้องการให้กับลูกค้า เพื่อส่งรายละเอียดไปยังขั้นตอนการจองหรือเช่ารถ
Method	ระบบจะเตรียมแคตตาล็อกออนไลน์ให้ลูกค้าเลือกตามที่ต้องการ หรือกรณีลูกค้าแบบ Walk-in ที่เข้ามาเพื่อจองหรือเช่ารถโดยตรงกับบริษัท ก็สามารถค้นหาได้จากแคตตาล็อกหนังสือที่ทางบริษัทได้จัดเตรียมไว้ให้