

Chapter 7

Process Model

(Structured System Approach)



ดร.สันติภูริ นรปิน

เรียบเรียง

อ.วไลลักษณ์ วงษ์รัตน์

Content

- 1) Process Model
- 2) System Model
- 3) Data Flow Diagram (DFD)

1 – Process Model



Process Model

- ❑ คือ แบบจำลองที่แสดงให้เห็นขั้นตอนการทำงานของระบบ
- ❑ มีวัตถุประสงค์เพื่อจำลองขั้นตอนการทำงานของระบบที่เป็นข้อความให้เป็นแผนภาพเพื่อความสะดวกในการสื่อสารระหว่าง SA กับผู้ที่เกี่ยวข้อง
- ❑ เครื่องมือที่ใช้ในขั้นตอนนี้ตามแนวทางเชิงโครงสร้างคือ แผนภาพกระแสข้อมูล (Data Flow Diagram : DFD)
- ❑ เครื่องมือที่ใช้ในขั้นตอนนี้ตามแนวทางเชิงวัตถุคือ Use Case Diagram ตามมาตรฐานของ UML

2 – System Model



แบบจำลองระบบสารสนเทศ

2.1 แบบจำลองเชิงตรรกะ (Logical Model)

2.2 แบบจำลองเชิงกายภาพ (Physical Model)

2.1 แบบจำลองเชิงตรรกะ (Logical Model)

- ❑ เป็นแบบจำลองที่อธิบายการดำเนินงานในระบบว่ามีการทำงานและความต้องการใดบ้างโดยไม่คำนึงถึงเทคโนโลยี หรือโปรแกรมภาษาใด ๆ ที่นำมาติดตั้งใช้งาน
- ❑ ไม่เน้นรายละเอียดทางเทคนิค

2.2 แบบจำลองเชิงกายภาพ (Physical Model)

- ❑ เป็นแบบจำลองที่นอกจากจะอธิบายการดำเนินงานของระบบว่าทำงานอะไรแล้ว ยังอธิบายว่ามีการดำเนินงานอย่างไร
- ❑ นอกจากนี้ยังมีการแสดงถึงประสิทธิภาพของเทคโนโลยีที่เลือกมาติดตั้งใช้งานเพื่อสนองความต้องการ และแสดงข้อจำกัดของเทคโนโลยีนั้น ๆ ด้วย
- ❑ เน้นรายละเอียดทางเทคนิค

3 – Data Flow Diagram



Data Flow Diagram

- 3.1 ความหมาย
- 3.2 วัตถุประสงค์
- 3.3 สัญลักษณ์ที่ใช้
- 3.4 แนวคิดของแบบจำลองการทำงานของระบบ
- 3.5 วิธีสร้างแบบจำลองขั้นตอนการทำงานของระบบด้วย DFD
- 3.6 แนวทางการสร้าง DFD ที่สมบูรณ์
- 3.7 Logical DFD และ Physical DFD
- 3.8 คำอธิบายการประมวลผล
- 3.9 พจนานุกรมข้อมูล

3.1 แผนภาพกระแสข้อมูล

- ❑ หมายถึง แผนภาพที่แสดงให้เห็นการเคลื่อนที่ของข้อมูลที่มีอยู่ในระบบ และการดำเนินงานที่เกิดขึ้นในระบบ
- ❑ แบ่งเป็น 2 ประเภทคือ Logical DFD และ Physical DFD
- ❑ ข้อมูลในแผนภาพทำให้ทราบถึง
 - ข้อมูลมาจากไหน, ข้อมูลไปที่ไหน, ข้อมูลเก็บที่ใด
 - เกิดเหตุการณ์ใดกับข้อมูลในระหว่างทาง
- ❑ แผนภาพกระแสข้อมูลจะแสดงภาพรวมของระบบ (Overall picture of a system) และรายละเอียดบางอย่าง


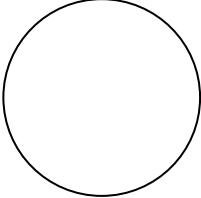
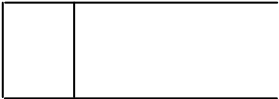



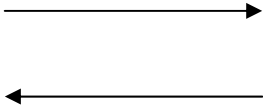
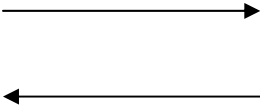
3.2 วัตถุประสงค์ของการสร้าง DFD

1. เป็นแผนภาพที่สรุปรวมข้อมูลทั้งหมดที่ได้จากการวิเคราะห์
ในลักษณะของรูปแบบที่เป็นโครงสร้าง
2. เป็นข้อตกลงร่วมกันระหว่างนักวิเคราะห์ระบบและผู้ใช้งาน
3. เป็นแผนภาพที่ใช้ในการพัฒนาต่อในขั้นตอนของการ
ออกแบบระบบ
4. เป็นแผนภาพที่ใช้ในการอ้างอิง หรือเพื่อใช้ในการพัฒนาต่อ
ในอนาคต
5. ทราบที่มาที่ไปของข้อมูลที่ไหลไปในกระบวนการต่าง ๆ
(Data and Process)

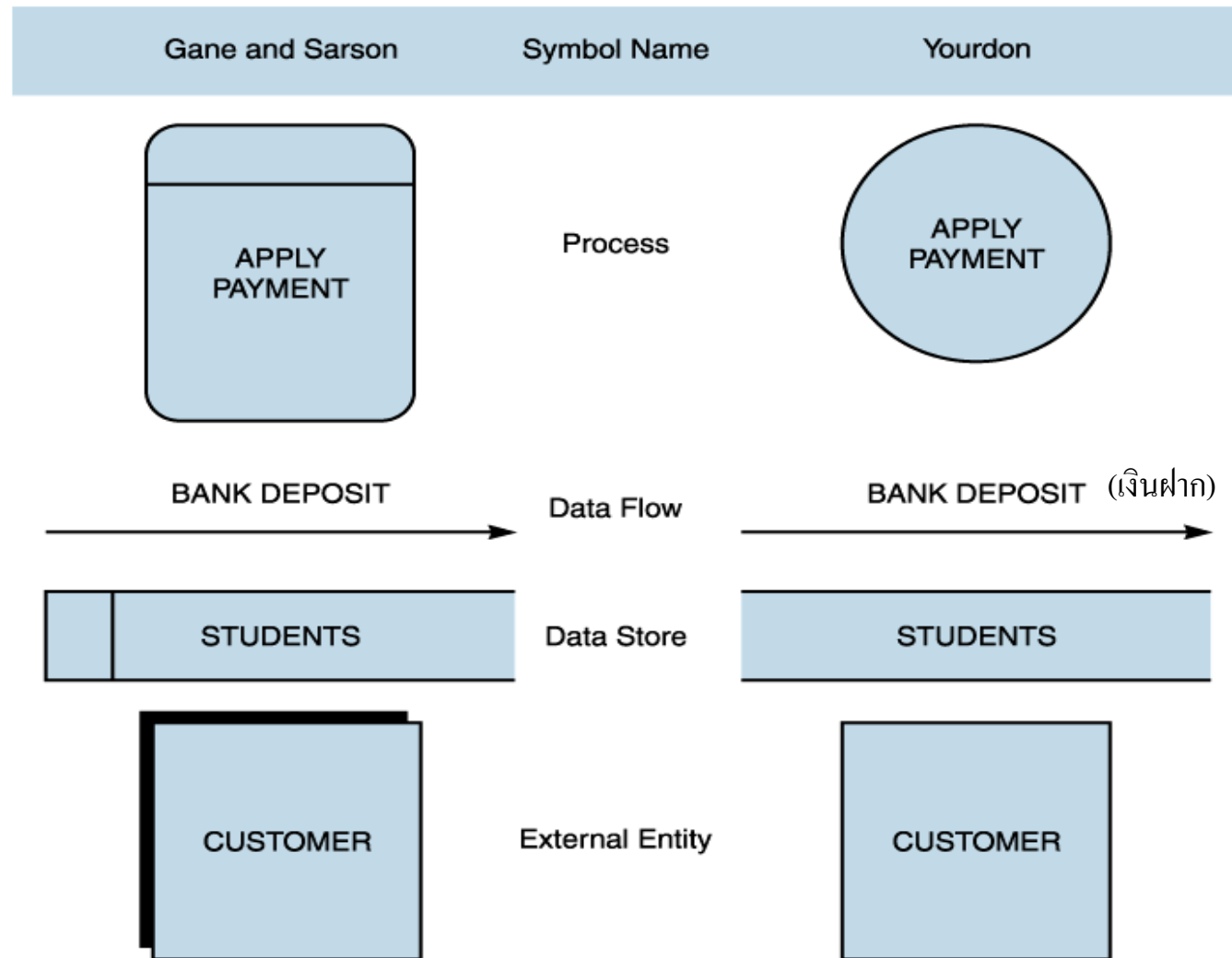
3.3 สัญลักษณ์ในแผนภาพกระแสข้อมูล

- ❑ สัญลักษณ์ที่ใช้เป็นมาตรฐานในการแสดงแผนภาพกระแสข้อมูลมีหลายชนิด
- ❑ ในที่นี้จะยกตัวอย่าง 2 ชนิด ได้แก่
 1. ชุดสัญญลักษณ์มาตรฐานที่พัฒนาโดย Gane and Sarson (1979)
 2. ชุดสัญญลักษณ์มาตรฐานที่พัฒนาโดย DeMarco and Yourdon (DeMarco, 1979); Yourdon and Constantine, 1979)

สัญลักษณ์ในแผนภาพกระแสข้อมูล

Gane & Sarson	DeMacro & Yourdon	ความหมาย
		Process : ขั้นตอนการทำงานภายในระบบ
		Data Store : แหล่งข้อมูลสามารถเป็นได้ทั้งไฟล์ข้อมูลและฐานข้อมูล (File or Database)
		External Entity : สัญลักษณ์แหล่งที่มาหรือปลายทางหรือสิ่งที่อยู่ภายนอกขอบเขตระบบ
		Data Flow : เส้นทางไหลของข้อมูลแสดงทิศทางของข้อมูลจากขั้นตอนการทำงานหนึ่งไปอีกขั้นตอนหนึ่ง

ตัวอย่างการใช้สัญลักษณ์ในแผนภาพกระแสข้อมูล



3.4 แนวคิดของแบบจำลองการทำงานของระบบ

3.4.1 ขั้นตอนการทำงานของระบบ (Process)

3.4.2 เส้นทางการไหลของข้อมูล (Data Flow)

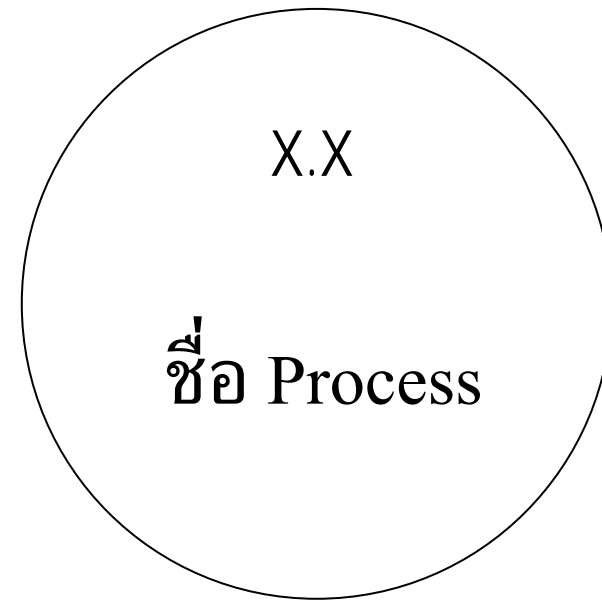
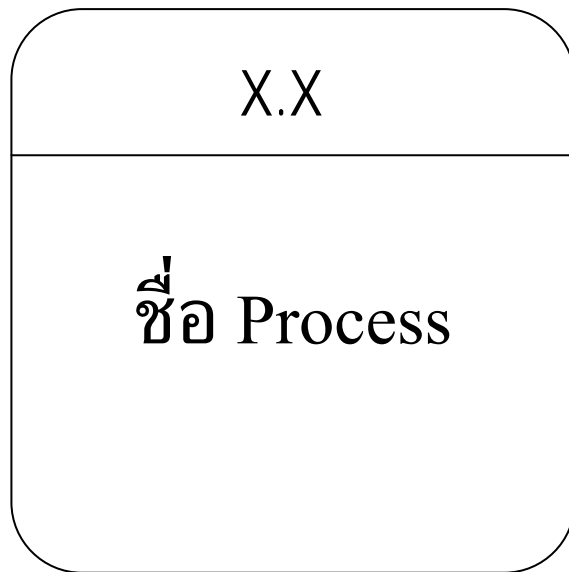
3.4.3 ตัวแทนข้อมูล (External Entity or External Agent)

3.4.4 แหล่งจัดเก็บข้อมูล (Data Store)

3.4.1 ขั้นตอนการทำงานของระบบ (Process)

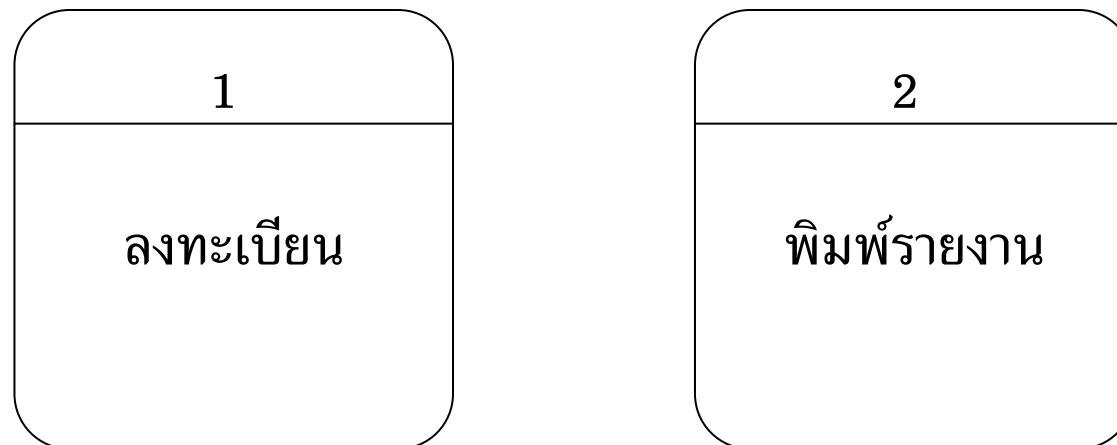
- ❑ คือ งานที่ดำเนินการ/ตอบสนองข้อมูลที่รับเข้า หรือ ดำเนินการ/ตอบสนองต่อเงื่อนไข/สถานะใด ๆ ที่เกิดขึ้น
- ❑ ขั้นตอนการดำเนินงานนั้นจะกระทำโดยบุคคล หน่วยงาน หุ่นยนต์ เครื่องจักร หรือ เครื่องคอมพิวเตอร์
- ❑ ชื่อโปรเซสจะเป็นกริยา (Verb) เช่น ลงทะเบียน เพิกถอนวิชา เพิ่มวิชา พิมพ์รายงาน เป็นต้น
- ❑ จำนวนโปรเซสควรมีอยู่ระหว่าง 2-7 โปรเซส หรือในบางตำรา ได้กำหนดจำนวนโปรเซสควรอยู่ในระหว่าง 7 ± 2

สัญลักษณ์ที่ใช้แสดงแทน Process

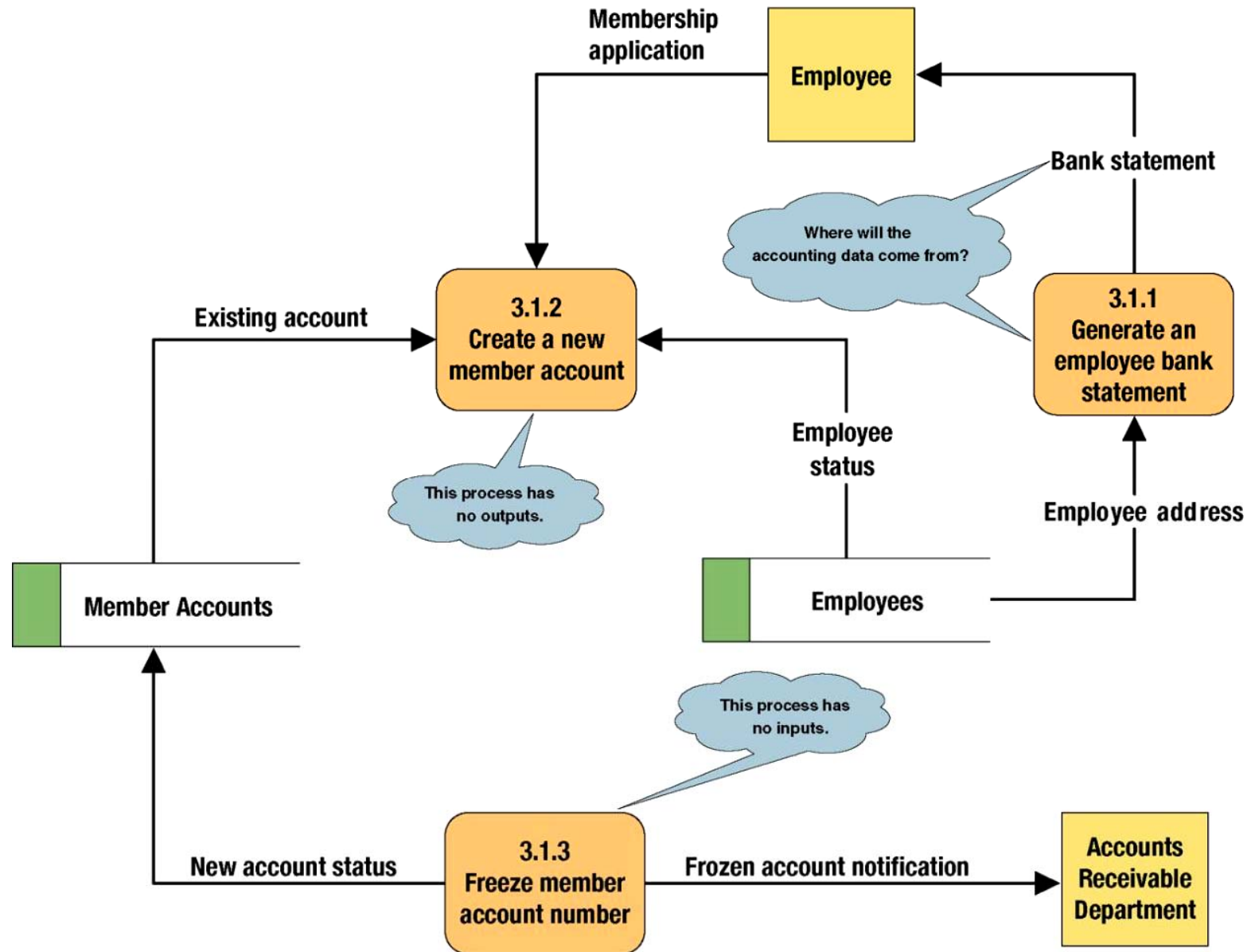


สัญลักษณ์ที่ใช้แสดงแทน Process (ต่อ)

- ❑ จากรูป แสดงสัญลักษณ์ที่ใช้แสดงแทน Process ด้วยสี่เหลี่ยมมุมมน ประกอบไปด้วย 2 ส่วน คือ ส่วนบนใช้แสดงหมายเลขของ Process เช่น 0, 1.0, 1.1 เป็นต้น ส่วนล่างจะใช้แสดงชื่อของ Process เช่น



ตัวอย่าง Process ที่ไม่ถูกต้อง



กฎของ Process

1. ต้องไม่มีข้อมูลรับเข้าเพียงอย่างเดียว

โดยไม่มีการส่งข้อมูลออกจากขั้นตอนการทำงาน (Process) เรียกข้อผิดพลาดชนิดนี้ว่า “Black Hole” เนื่องจากข้อมูลที่รับเข้ามาแล้วสูญหายไป จากรูป คือ Process 3.1.2 ที่มีข้อผิดพลาดลักษณะนี้

2. ต้องไม่มีข้อมูลออกเพียงอย่างเดียว

โดยไม่มีข้อมูลเข้าสู่ Process เลย จากรูป คือ Process 3.1.3 ที่มีข้อผิดพลาดลักษณะนี้

กฎของ Process (ต่อ)

3. ข้อมูลรับเข้าจะต้องเพียงพอในการสร้างข้อมูลส่งออก
กรณีที่มีข้อมูลที่รับเข้าไม่เพียงพอในการสร้างข้อมูลส่งออก เรียกว่า “Gray Hole” โดยอาจเกิดจากการรวบรวม ข้อเท็จจริงและข้อมูลไม่สมบูรณ์ หรือการใช้ชื่อข้อมูลรับเข้า และข้อมูลส่งออกผิดจากรูปคือ Process 3.1.1 ที่มี ข้อผิดพลาดลักษณะเช่นนี้ เนื่องจากข้อมูลที่รับเข้ามามีเพียง ที่อยู่ของพนักงาน (Employee Address) แต่ไม่มีข้อมูล กระแสเงินสดในธนาคารของลูกค้าที่เข้าสู่ Process ดังนั้น ข้อมูลจึงไม่เพียงพอที่จะสร้างเป็นรายงานสถานะทางการเงิน ทางธนาคารของพนักงานได้ (Bank Statement)

กฎของ Process (ต่อ)

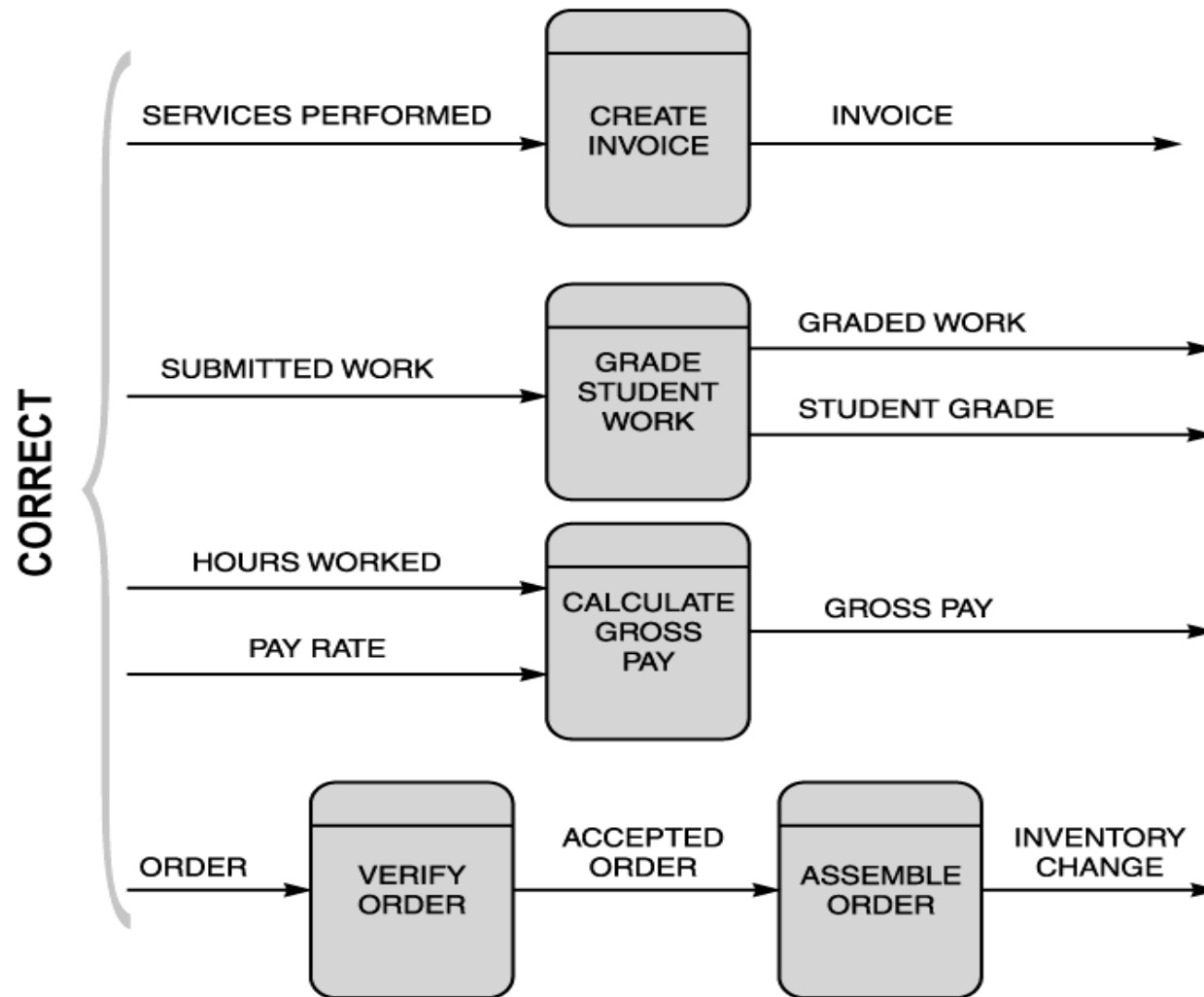
4. การตั้งชื่อ Process ต้องใช้คำกริยา (Verb)

เช่น Prepare Management Report, Calculate Data สำหรับ
ภาษาไทยใช้เป็นคำกริยาเช่นเดียวกัน เช่น บันทึกข้อมูลใบสั่ง
ซื้อ ตรวจสอบข้อมูลลูกค้า คำนวณเงินเดือน เป็นต้น

ลักษณะทั่วไปของโปรเซสที่เกี่ยวข้องกับการประมวลผลข้อมูล

- ❑ บันทึกลง (Inserts)
- ❑ แสดง (Displays)
- ❑ ปรับปรุง (Update)
- ❑ ลบ (Delete)
- ❑ คำนวณ (Compute)

ตัวอย่าง Process ที่ถูกต้อง



3.4.2 เส้นทางไหลของข้อมูล (Data Flow)

- เป็นการสื่อสารระหว่าง Process ต่างๆ และสภาพแวดล้อมภายนอกหรือภายในระบบ โดยแสดงถึงข้อมูลที่นำเข้าไปในแต่ละ Process และข้อมูลที่ส่งออกจาก Process ใช้ในการแสดงถึงการบันทึกข้อมูล การลบข้อมูล การแก้ไขข้อมูลต่างๆ ในไฟล์หรือในฐานข้อมูล ซึ่งใน Data Flow Diagram เรียกว่า “Data Store”
- สัญลักษณ์ที่ใช้อธิบายเส้นทางไหลของข้อมูลคือ เส้นตรงที่ประกอบด้วยหัวลูกศรตรงปลายเพื่อบอกทิศทางการเดินทางหรือการไหลของข้อมูล



กฎของ Data Flow

1. ชื่อของ Data Flow ควรเป็นชื่อของข้อมูลที่ส่งโดยไม่ต้องอธิบายว่าส่งอย่างไร ทำงานอย่างไร
2. Data Flow ต้องมีจุดเริ่มต้นหรือสิ้นสุดที่ Process เพราะ Data Flow คือข้อมูลนำเข้า (Inputs) และข้อมูลส่งออก (Outputs) ของ Process
3. Data Flow จะเดินทางระหว่าง External Entity กับ External Entity ไม่ได้
4. Data Flow จะเดินทางจาก External Entity ไป Data Store ไม่ได้

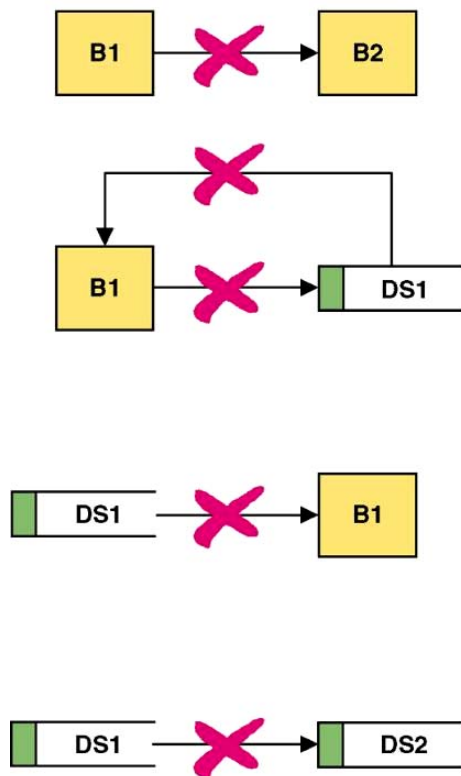
กฎของ Data Flow (ต่อ)

5. Data Flow จะเดินทางจาก Data Store ไป External Entity ไม่ได้
6. Data Flow จะเดินทางระหว่าง Data Store กับ Data Store ไม่ได้
7. การตั้งชื่อ Data Flow จะต้องใช้คำนาม (Noun) เช่น Inventory Data, Goods Sold Data เป็นต้น

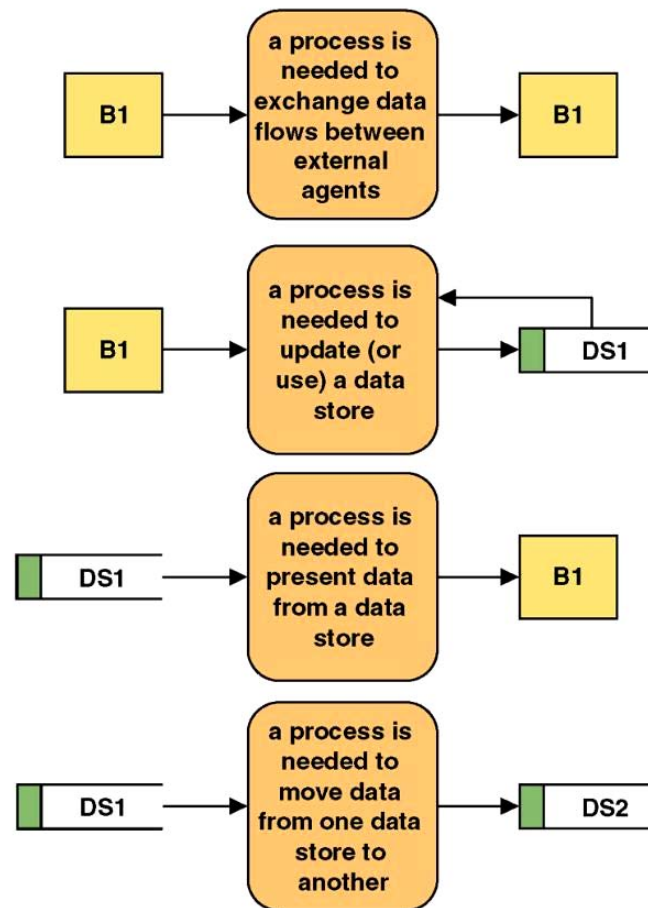
Illegal Data Flows



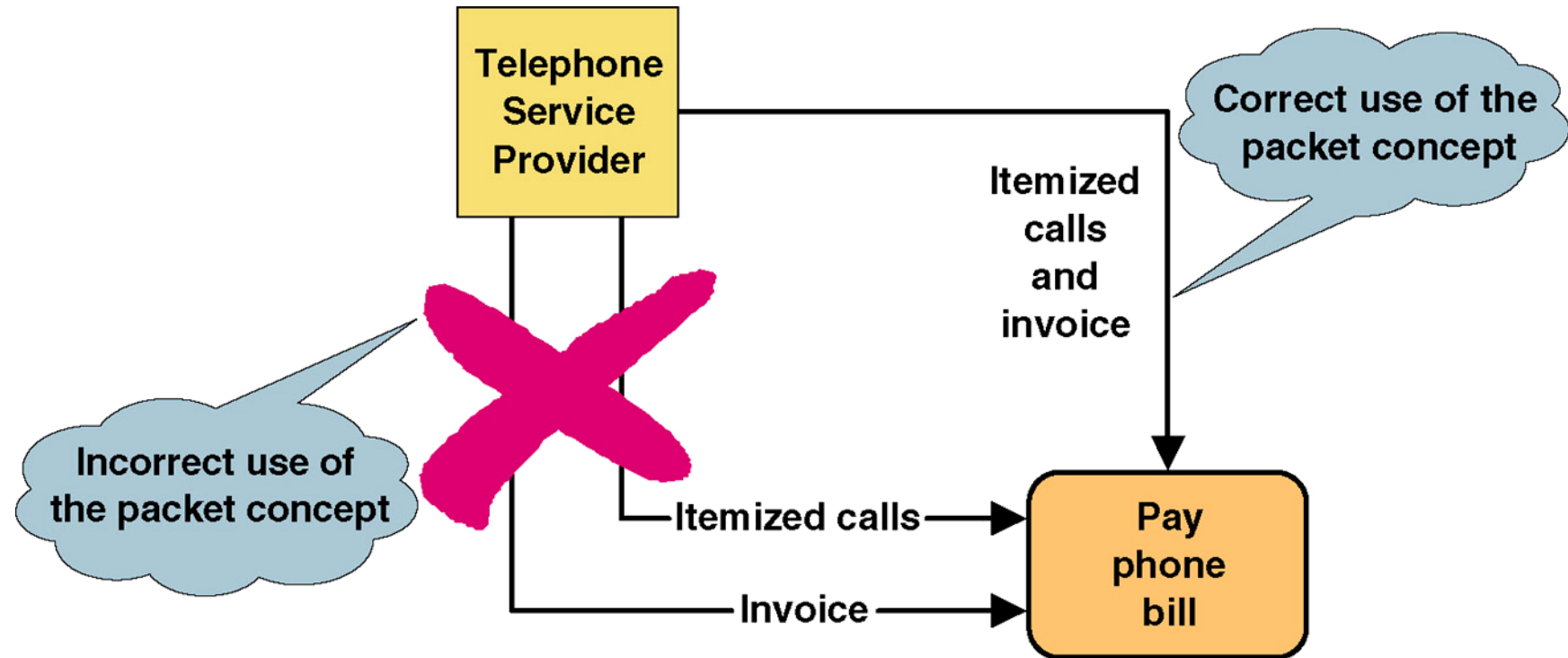
Illegal
data
flows



Corrected
data
flows



Data Flow Packet Concept



3.4.3 เอนทิตีภายนอก (External Entity)

- หมายถึง บุคคล หน่วยงานในองค์กร องค์กรอื่น ๆ หรือระบบงานอื่น ๆ ที่อยู่ภายนอกขอบเขตของระบบ แต่มีความสัมพันธ์กับระบบ โดยมีการส่งข้อมูลเข้าสู่ระบบเพื่อดำเนินงาน และรับข้อมูลที่ผ่านการดำเนินงานเรียบร้อยแล้วจากระบบ ในบางครั้งเรียกว่า “External Agent” หรือ “Source/Sink”
- สัญลักษณ์ที่ใช้อธิบาย คือ สีเหลี่ยมจัตุรัส หรือสีเหลี่ยมผืนผ้า ภายในจะต้องแสดงชื่อของ External Entity โดยสามารถทำการซ้ำ (Duplicate) ได้ด้วยการใช้เครื่องหมาย \ (back slash) ตรงมุมล่างซ้าย

สัญลักษณ์ของ External Entity

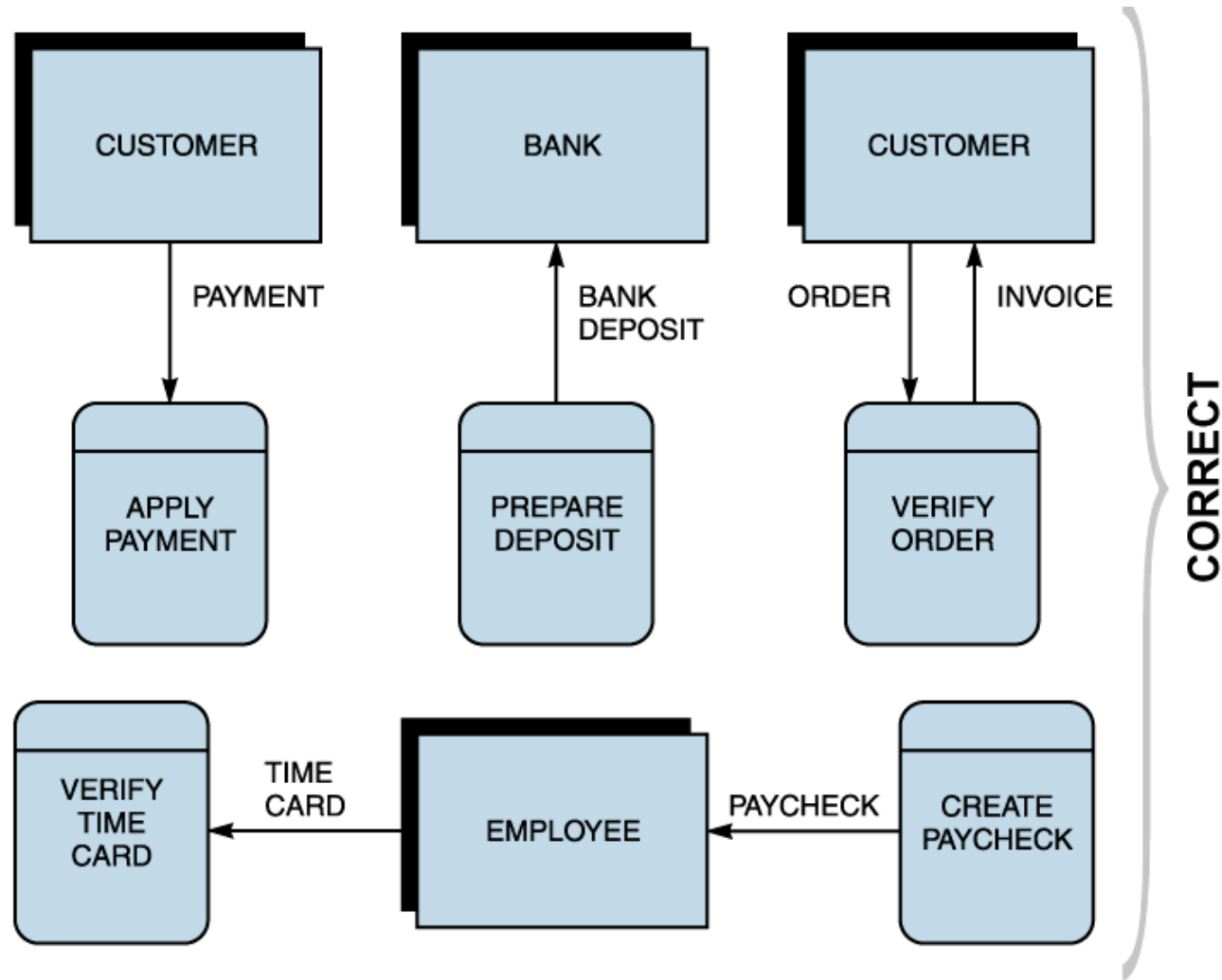
ชื่อ External Entity

ชื่อ External Entity

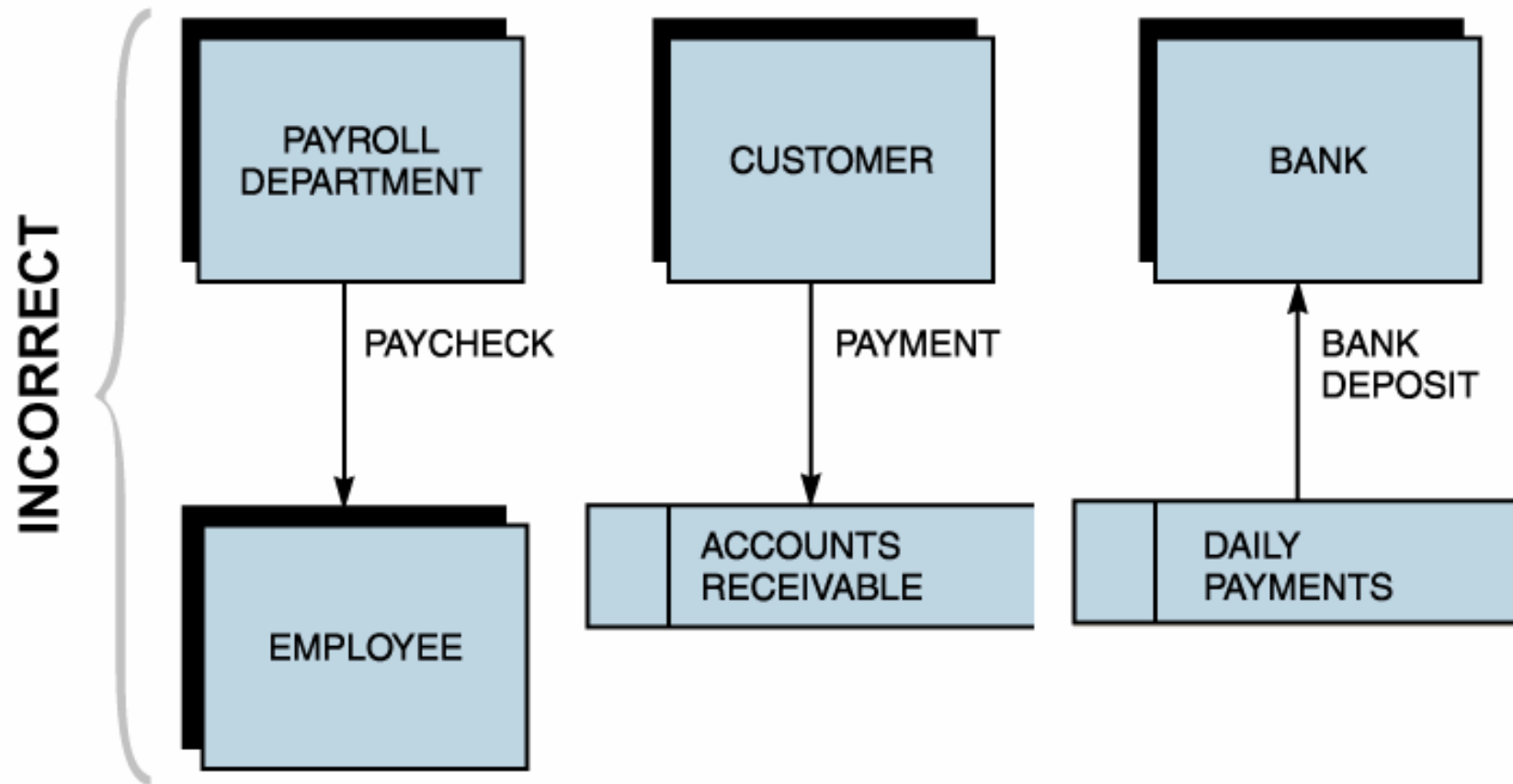
กฎของ External Entity

1. ข้อมูลจาก External Entity จะวิ่งไปสู่อีก External Entity หนึ่งโดยตรงไม่ได้ จะต้องผ่าน Process ก่อนเพื่อประมวลข้อมูลนั้น จึงได้ข้อมูลออกไปสู่อีก External Entity
2. การตั้งชื่อ External Entity ต้องใช้คำนาม (Noun) เช่น Customer, Bank เป็นต้น

ตัวอย่างการใช้ External Entity ที่ถูกต้อง



ตัวอย่างการใช้ External Entity ที่ไม่ถูกต้อง



3.4.4 แหล่งจัดเก็บข้อมูล (Data Store)

- เป็นแหล่งเก็บ/บันทึกข้อมูล เปรียบเสมือนคลังข้อมูล
- เทียบเท่ากับไฟล์ข้อมูล และฐานข้อมูล โดยอธิบายรายละเอียดและคุณสมบัติเฉพาะตัวของสิ่งที่ต้องการเก็บ/บันทึก

สัญลักษณ์ของ Data Store

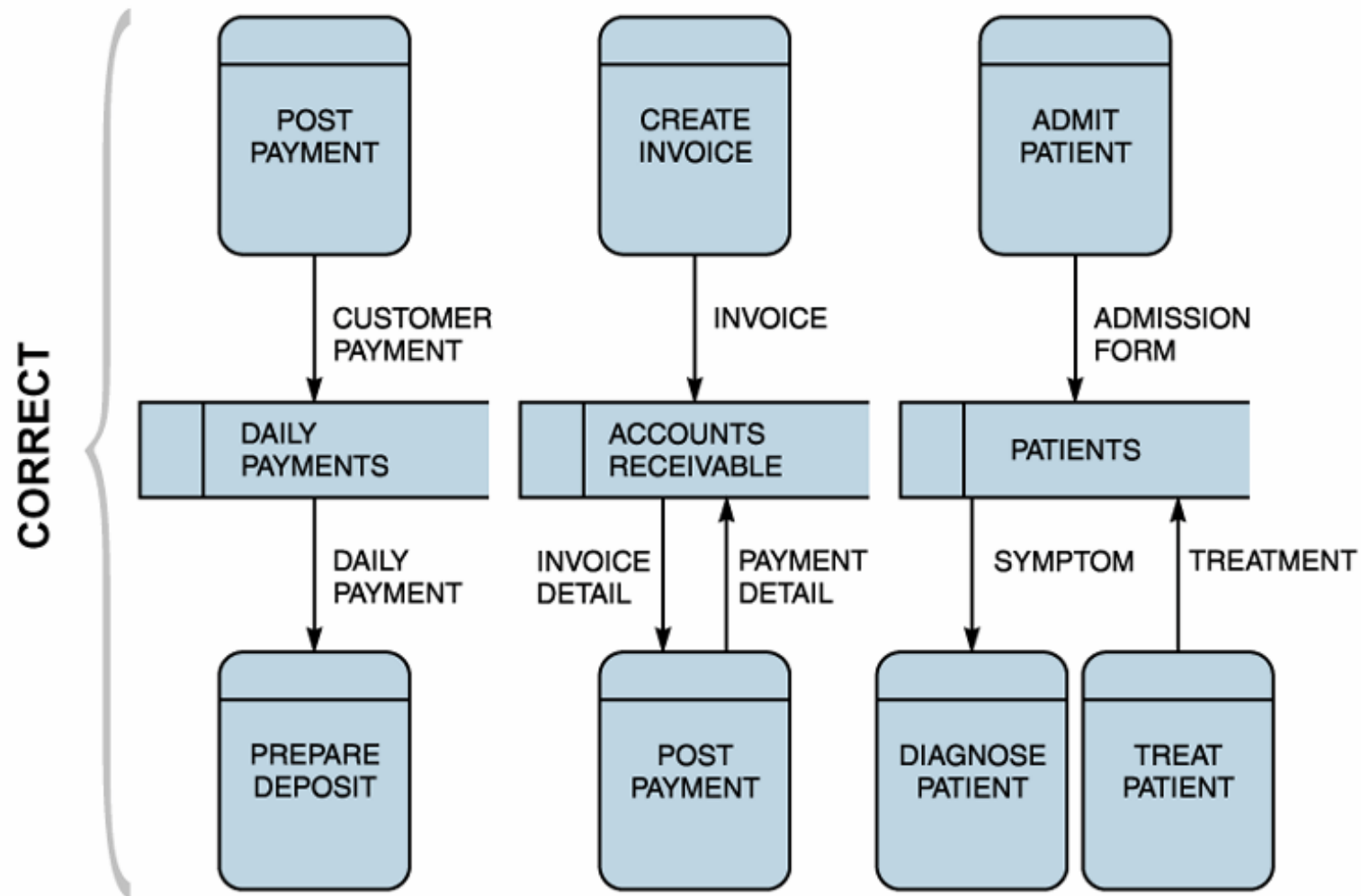
- สัญลักษณ์ที่ใช้อธิบายคือสี่เหลี่ยมเปิดหนึ่งข้าง แบ่งออกเป็นสองส่วน ได้แก่
 - ส่วนที่ 1 ทางด้านซ้ายใช้แสดงรหัสของ Data Store อาจจะเป็นหมายเลขลำดับหรือตัวอักษรได้เช่น D1, D2 เป็นต้น
 - ส่วนที่ 2 ทางด้านขวา ใช้แสดงชื่อ Data Store หรือชื่อไฟล์ เช่น Employee, Application, Member เป็นต้น

รหัส	ชื่อ Data Store	ชื่อ Data Store
------	-----------------	-----------------

กฎของ Data Store

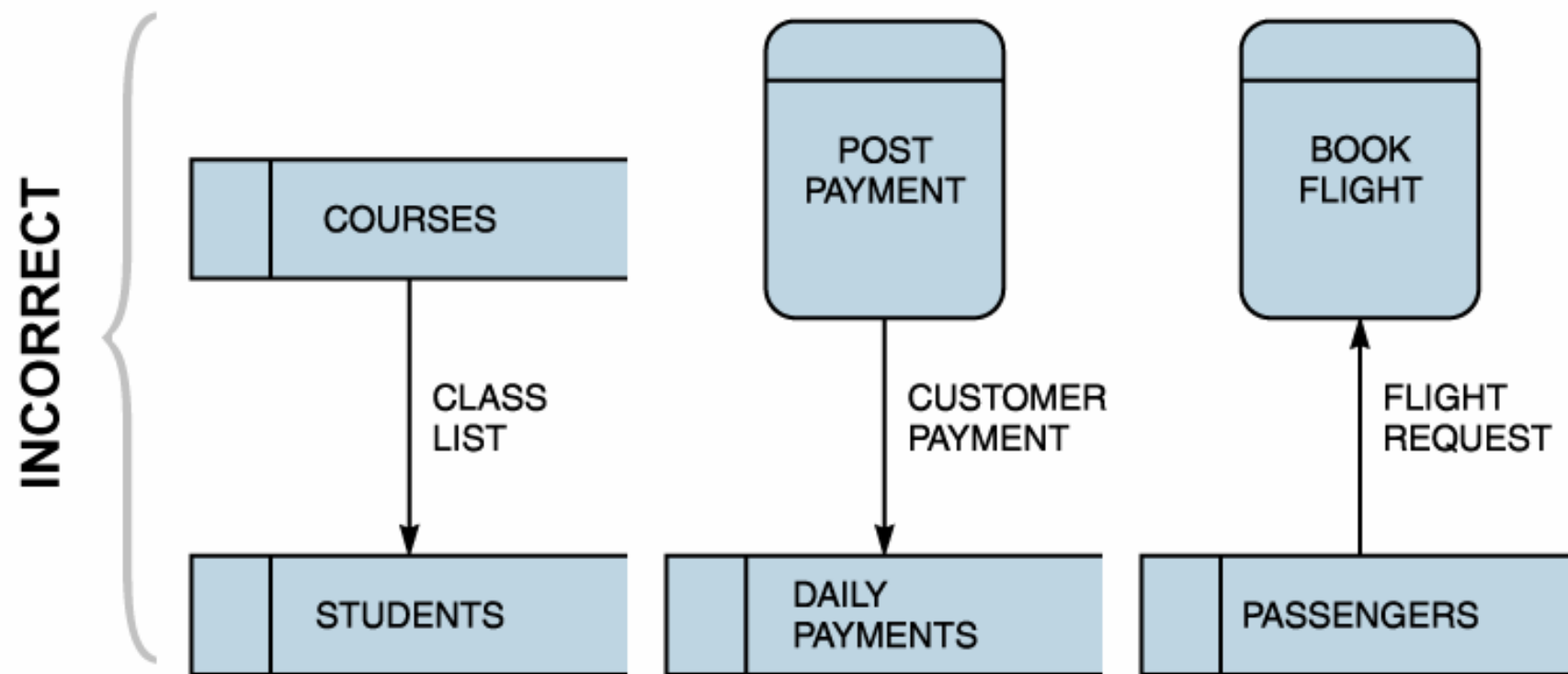
1. ข้อมูลจาก Data Store หนึ่งจะวิ่งไปสู่อีก Data Store หนึ่ง โดยตรงไม่ได้ จะต้องผ่านการประมวลผลจาก Process ก่อน
2. ข้อมูลจาก External Entity จะวิ่งเข้าสู่ External Entity โดยตรงไม่ได้
3. การตั้งชื่อ Data Store จะต้องใช้คำนาม (Noun) เช่น Customer File, Inventory หรือ Employee File เป็นต้น

ตัวอย่างของ Process, Data Flow และ Data Store ที่ถูกต้อง



วินิจฉัย





ตัวอย่างของ Process, Data Flow และ Data Store ที่ไม่ถูกต้อง



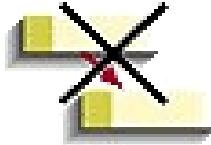


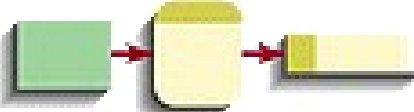
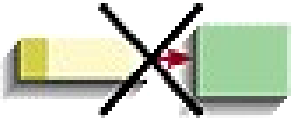
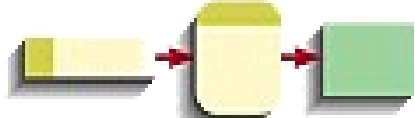
สรุปการใช้สัญลักษณ์ต่าง ๆ ภายใน DFD

Data Flow เชื่อมต่อกับ	ได้	ไม่ได้
โพรเซสหนึ่ง ไปยังอีก โพรเซสหนึ่ง	✓	
โพรเซส ไปยัง เอนทิตีภายนอก	✓	
โพรเซส ไปยัง ดาต้าสโตร์	✓	
เอนทิตีหนึ่ง ไปยังอีก เอนทิตีหนึ่ง		✓
เอนทิตี ไปยัง ดาต้าสโตร์		✓
ดาต้าสโตร์หนึ่ง ไปยังอีก ดาต้าสโตร์หนึ่ง		✓



Data Flow Diagramming Rules : Process

	Rule	Incorrect	Correct
A.	No process can have only outputs (a miracle)		
B.	No process can have only inputs (black hole)		
C.	A process has a verb phrase label (except for context diagram)		

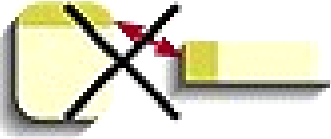

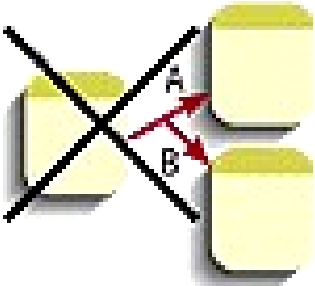
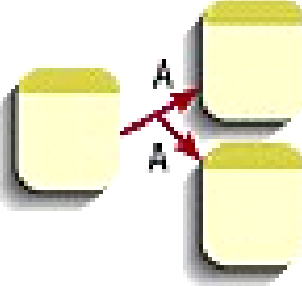
Data Flow Diagramming Rules : Data Store

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

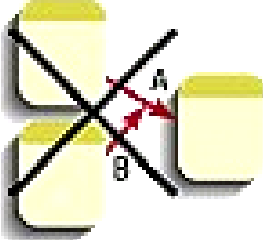
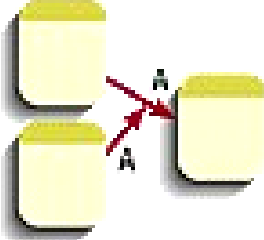
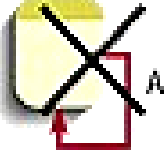
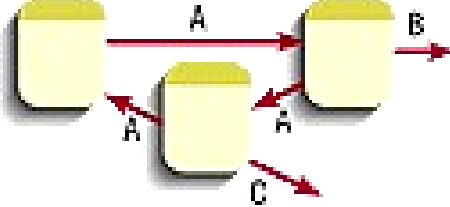
Data Flow Diagramming Rules : Source/Sink

	Rule	Incorrect	Correct
H.	Data cannot move directly from a source to a sink		
I.	A source/sink has a noun phrase label		

Data Flow Diagramming Rules : Data Flow

	Rule	Incorrect	Correct
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 sources/sinks		

Data Flow Diagramming Rules : Data Flow (Cont.)

	Rule	Incorrect	Correct
L.	A join means that exactly the same data come from any two or more different processes, data stores or sources/sinks to a common location		
M.	A data flow cannot go directly back to the same process it leaves		
N.	A data flow to a data store means update (delete or change)		
O.	A data flow from a data store means retrieve or use		
P.	Data flow has a noun phrase label		

3.5 วิธีสร้างแบบจำลองขั้นตอนการทำงานของระบบ ด้วย DFD มีหลักเกณฑ์ดังนี้

1. แต่ละ Context Diagram ต้องสมมูลอยู่ภายในหนึ่งหน้ากระดาษ
2. ชื่อของกระบวนการใน Context Diagram ควรเป็นชื่อของระบบงานหรือโครงการ
3. ให้ใช้ชื่อเดียวกัน ในเรื่องเดียวกันตลอดทั้งระบบ
4. ไม่ควรลากเส้นตัดกัน
5. หมายเลขอ้างอิงในแต่ละสัญลักษณ์ของกระบวนการ ต้องไม่ซ้ำกัน

วิธีสร้างแบบจำลองขั้นตอนการทำงานของระบบด้วย DFD

3.5.1 สร้างแผนภาพบริบท (Context Diagram)

3.5.2 สร้างแผนภาพระดับ 0 (Level-0 Diagram)

3.5.3 แบ่งย่อยแผนภาพ (Decomposition of DFD)

3.5.4 ตรวจสอบสมดุลของ DFD (Balancing DFD)

3.5.1 สร้างแผนภาพบริบท (Context Diagram)

- ❑ แผนภาพบริบท (Context Diagram) คือ แผนภาพกระแสข้อมูลระดับบนสุดที่แสดงภาพรวมการทำงานของระบบที่มีความสัมพันธ์กับสภาพแวดล้อมภายนอก
- ❑ แผนภาพบริบทแสดงให้เห็นขอบเขต และเส้นแบ่งเขตของระบบที่ศึกษาและพัฒนา
- ❑ การสร้างแบบจำลองขั้นตอนการทำงานของระบบ นักวิเคราะห์ระบบควรทำการสร้าง Context Diagram ก่อน

แนวทางในการกำหนดขอบเขต

1. เปรียบระบบเสมือนภาษนะบรรจุ เพื่อแบ่งแยกสิ่งที่อยู่ภายในภาษนะออกจากสิ่งที่อยู่ภายนอกภาษนะ โดยไม่ต้องสนใจสิ่งที่อยู่ภายในภาษนะมีอะไรบ้าง
2. ศึกษากระบวนการโดยอาจจะการสอบถามผู้ใช้งานถึงเหตุการณ์ (Event) หรือ การดำเนินงานประจำวันที่เกิดขึ้นของระบบว่ามีการติดต่อ จัดการ หรือดำเนินงานอย่างไรบ้าง และระบบมีการตอบสนองต่อเหตุการณ์นั้นๆ อย่างไร อะไรคือข้อมูลที่รับเข้ามา (Input) และส่งมาจากใคร (External Entity)

แนวทางในการกำหนดขอบเขต (ต่อ)

3. สอบถามผู้ใช้ระบบว่าระบบจะต้องส่งข้อมูลอะไร (Output) ออกไปสู่ External Entity บ้าง ต้องการรูปแบบรายงาน การสอบถามข้อมูล (Query) แบบใด สิ่งเหล่านี้ทำให้ SA สามารถพิจารณาการวาด Data Flow ได้
4. จำแนกแหล่งข้อมูลภายนอกระบบ (External data store) ที่ระบบต้องการจากไฟล์หรือฐานข้อมูลจากระบบอื่น ซึ่งอาจเป็นการอ่าน แก้ไข เปลี่ยนแปลง ข้อมูลเหล่านั้น
5. ทำการวาด Context Diagram จากสิ่งที่รวบรวมได้จากข้อ 1-4

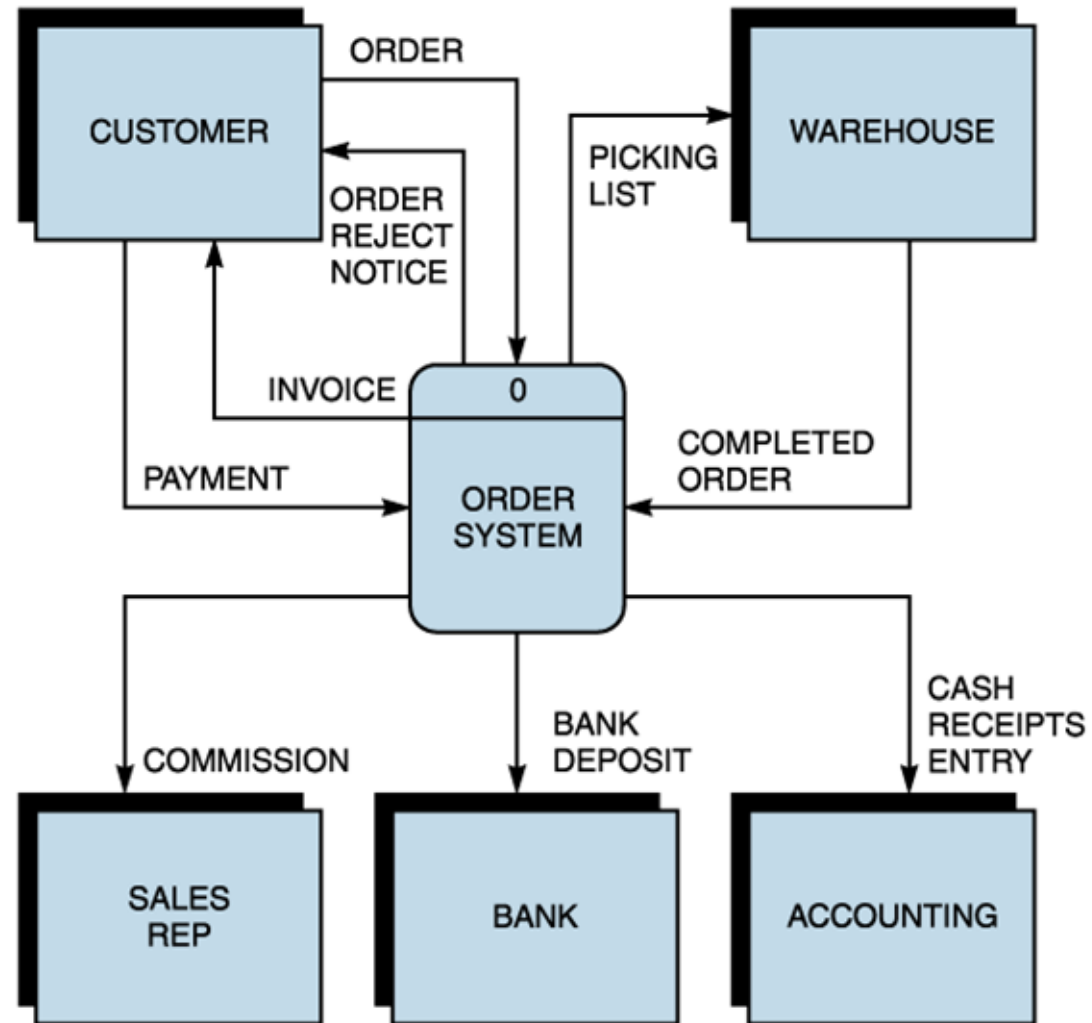
การกำหนดรายละเอียดในระบบงาน

- ❑ หลังจากที่ได้ศึกษาการทำงาน ข้อมูลรับเข้า ข้อมูลส่งออก SA อาจมีเส้นทางการไหลของข้อมูล (Data Flow) มากมาย ซึ่งไม่อาจแสดงได้ทั้งหมดใน Context Diagram นี้
- ❑ ดังนั้น Data Flow ที่แสดงใน Context Diagram ควรเป็นข้อมูลหลักและมีความสำคัญต่อระบบ ส่วนรายละเอียดของการเคลื่อนไหวของข้อมูลนั้นสามารถนำไปอธิบายใน DFD ระดับต่อไปได้

สิ่งที่ปรากฏใน Context Diagram

- ❑ ใน Context Diagram ประกอบด้วย Process ที่แทน Process ของระบบทั้งหมดเพียงหนึ่ง Process เท่านั้นที่อยู่ภายในขอบเขตของระบบ
- ❑ ให้แสดงหมายเลขศูนย์ (“0”) ตรงส่วนบนของสัญลักษณ์ Process
- ❑ ใน Context Diagram ยังแสดงรายละเอียดของ External Entity และ External Data Store รอบ ๆ ขั้นตอนการดำเนินงาน (ภายนอกขอบเขตของระบบ) และมี Data Flows แสดงการติดต่อระหว่างระบบกับสิ่งที่อยู่ภายนอก
- ❑ ภายใน Context Diagram จะต้องไม่มี Data Store ปรากฏอยู่

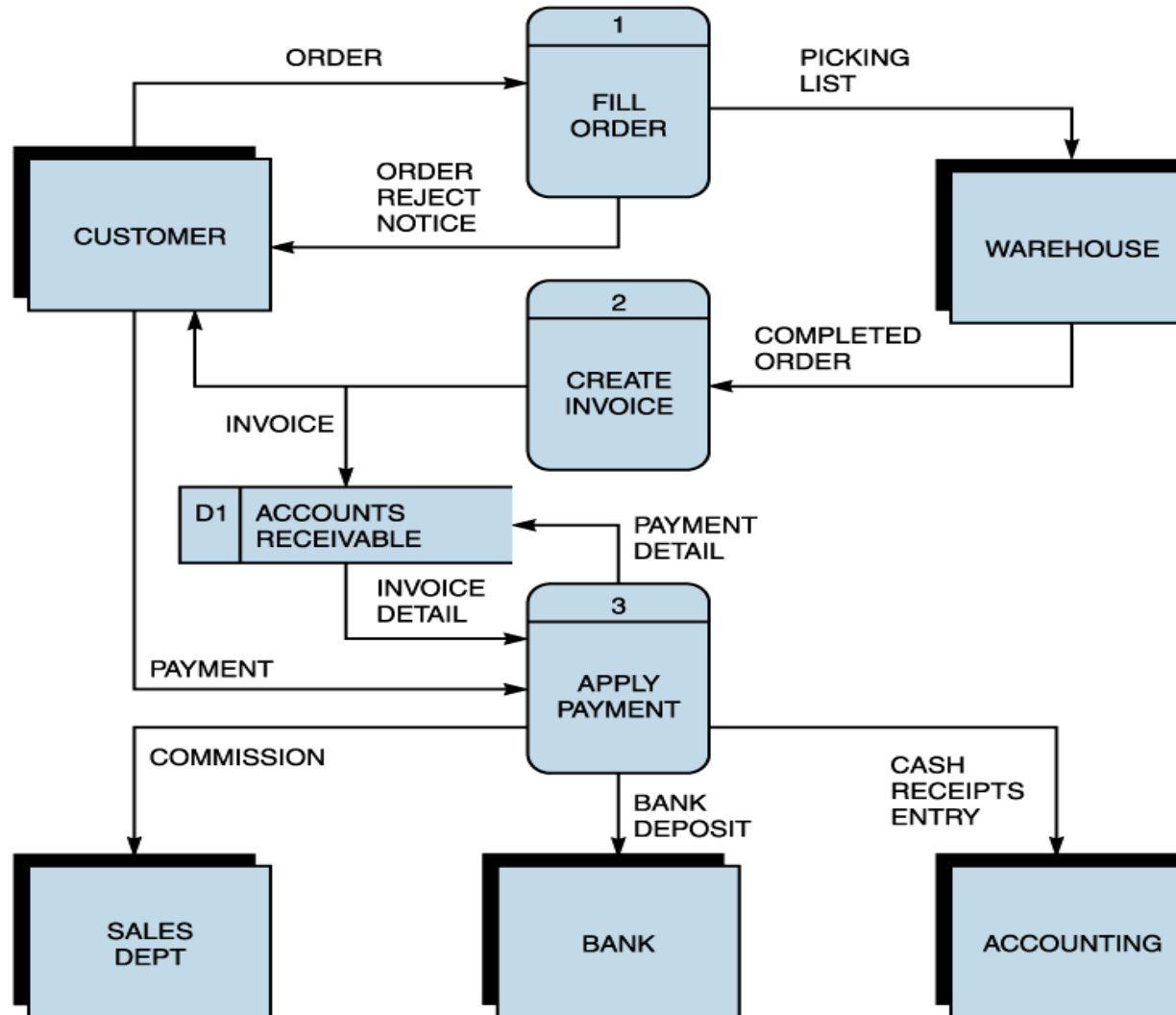
Example : Context Diagram of Order System



3.5.2 สร้างแผนภาพระดับ 0 (Level-0 Diagram)

- ❑ เป็นแผนภาพกระแสข้อมูลในระดับที่แสดงขั้นตอนการทำงานหลักทั้งหมด (Process หลัก) ของระบบ
- ❑ Level-0 Diagram จะแสดงทิศทางการไหลของ Data Flow และแสดงรายละเอียดของแหล่งจัดเก็บข้อมูล (Data Store)
- ❑ เป็นการแสดงให้เห็นถึงรายละเอียดของ Process การทำงานหลัก ๆ ที่มีอยู่ภายในภาพรวมของระบบ (Context Diagram) ว่ามีขั้นตอนใดบ้าง
- ❑ แต่ละ Process จะมีหมายเลขกำกับอยู่ด้านบนของสัญลักษณ์ตั้งแต่ 1 เป็นต้นไป

Example : Level-0 DFD of Order System



3.5.3 แบ่งย่อยแผนภาพ (Decomposition of DFD)

- ❑ ถ้าระบบใดมีการทำงานที่ซับซ้อนมาก SA จะไม่สามารถอธิบายการทำงานทั้งหมดได้ภายในขั้นตอนเดียวใน Context Diagram
- ❑ ดังนั้นในการวิเคราะห์ระบบจึงสามารถจำแนกระบบใหญ่หนึ่งระบบออกเป็นระบบย่อยๆ ได้หลายระบบ โดยแบ่งให้เป็นระบบย่อยที่มีขนาดเล็กลงเรื่อย ๆ จนสามารถอธิบายการทำงานได้ทั้งหมด เรียกวิธีนี้ว่า “การแบ่งย่อย (Decomposition) หรือ Functional Decomposition”

แบ่งย่อยแผนภาพ (ต่อ)

- ❑ Decomposition คือ การแบ่ง/แยก/ย่อยระบบและขั้นตอนการทำงานออกเป็นส่วนย่อย โดยในแต่ละขั้นตอนที่แยกออกมา (Subsystems) จะแสดงให้เห็นถึงรายละเอียดของการทำงานเพิ่มมากขึ้น
- ❑ การแบ่งย่อย Process นั้นสามารถแบ่งย่อยลงไปได้เรื่อย ๆ จนกระทั่งถึงระดับที่ไม่สามารถแบ่งย่อยได้อีกแล้ว

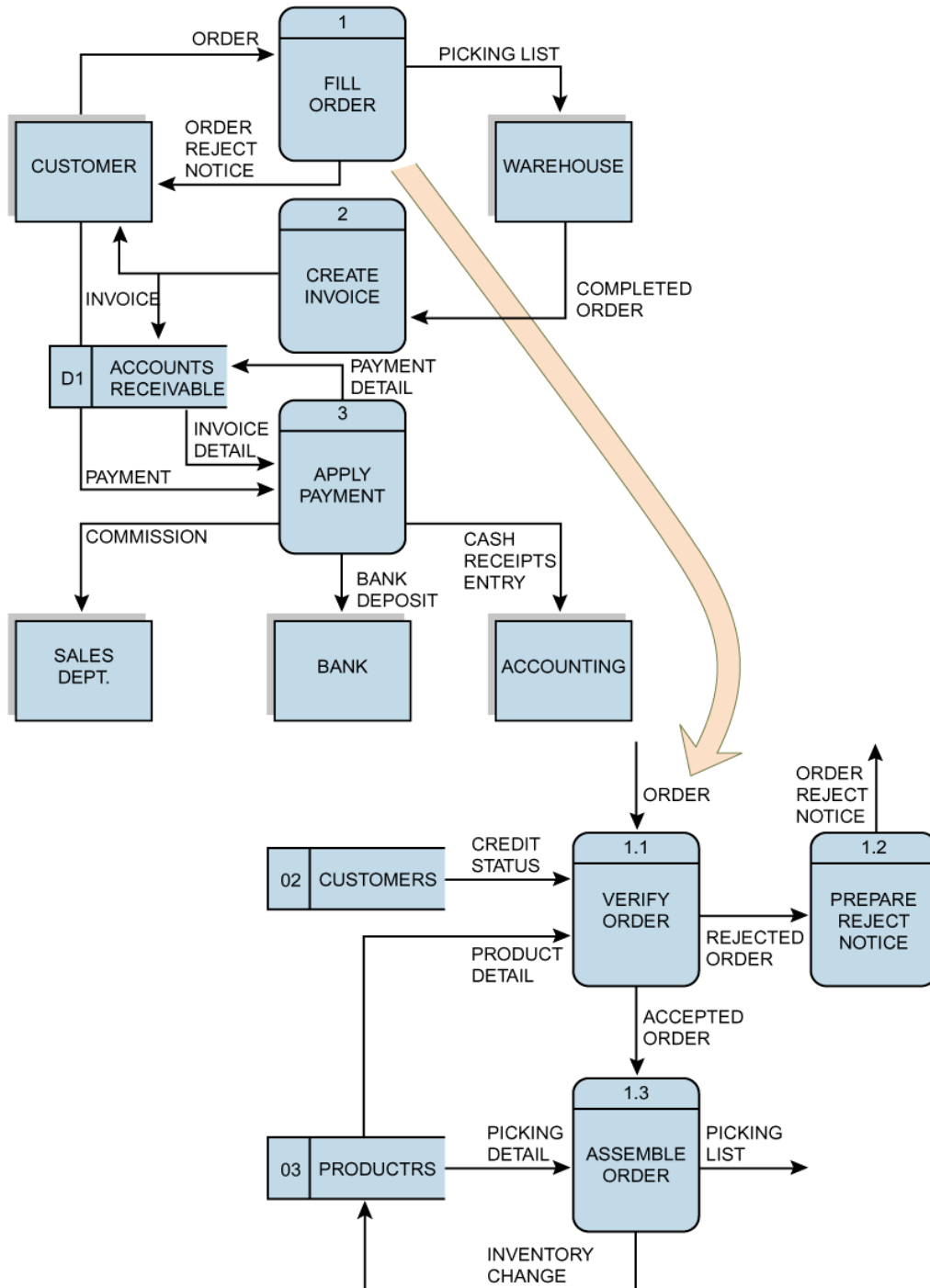
แบ่งย่อยแผนภาพ (ต่อ)

- ❑ แผนภาพที่ไม่สามารถแบ่งย่อย Process ได้อีกแล้วว่า Primitive DFD
- ❑ ระดับของแผนภาพที่แบ่งย่อยมาจาก Level-0 เรียกว่า Level-1
- ❑ แผนภาพที่แบ่งย่อยในระดับถัดมาจาก Level-0 diagram จะต้อง มี Process อย่างน้อย 2 Process ขึ้นไป

Example :

Order System

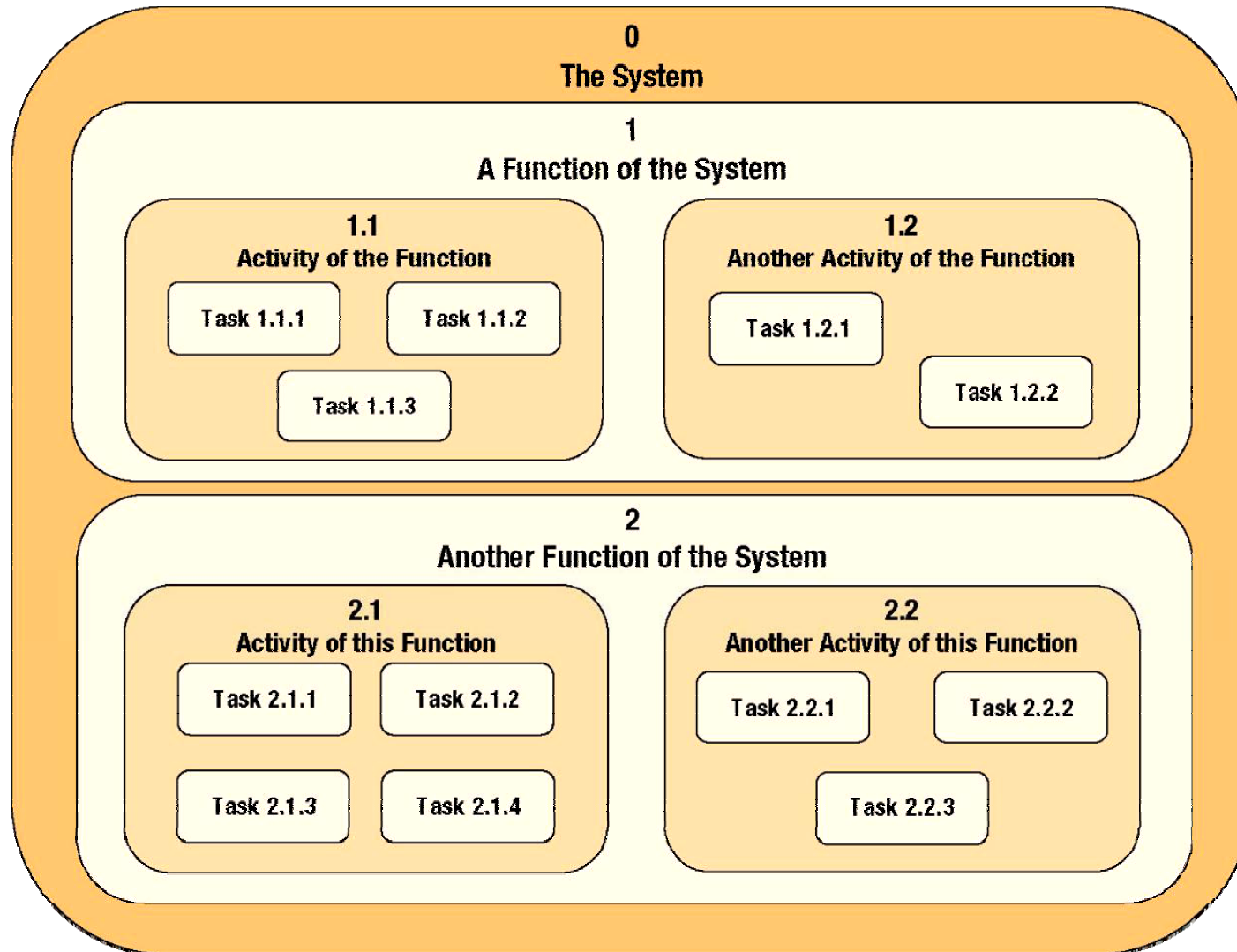
Decomposition of DFD



Functional Decomposition Diagram

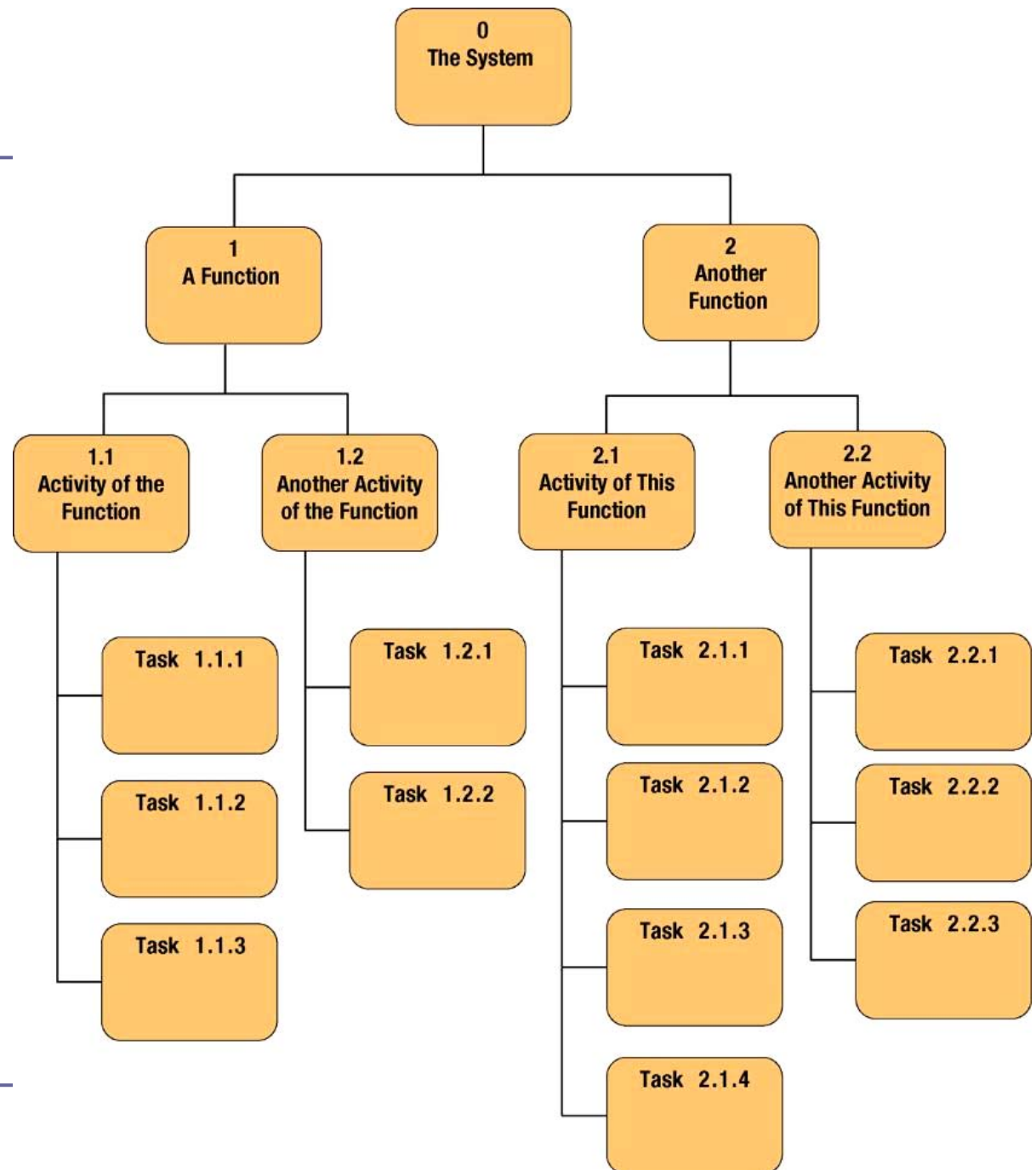
- ❑ คือรูปภาพที่ใช้แทนการแตกย่อยของระบบออกเป็นระบบย่อย (Subsystem) หรือ ฟังก์ชัน (Function) ย่อย ลงไปตามลำดับ
- ❑ Functional Decomposition Diagram สามารถเรียกอีกอย่างหนึ่งได้ว่า Hierarchy Chart

Process Decomposition

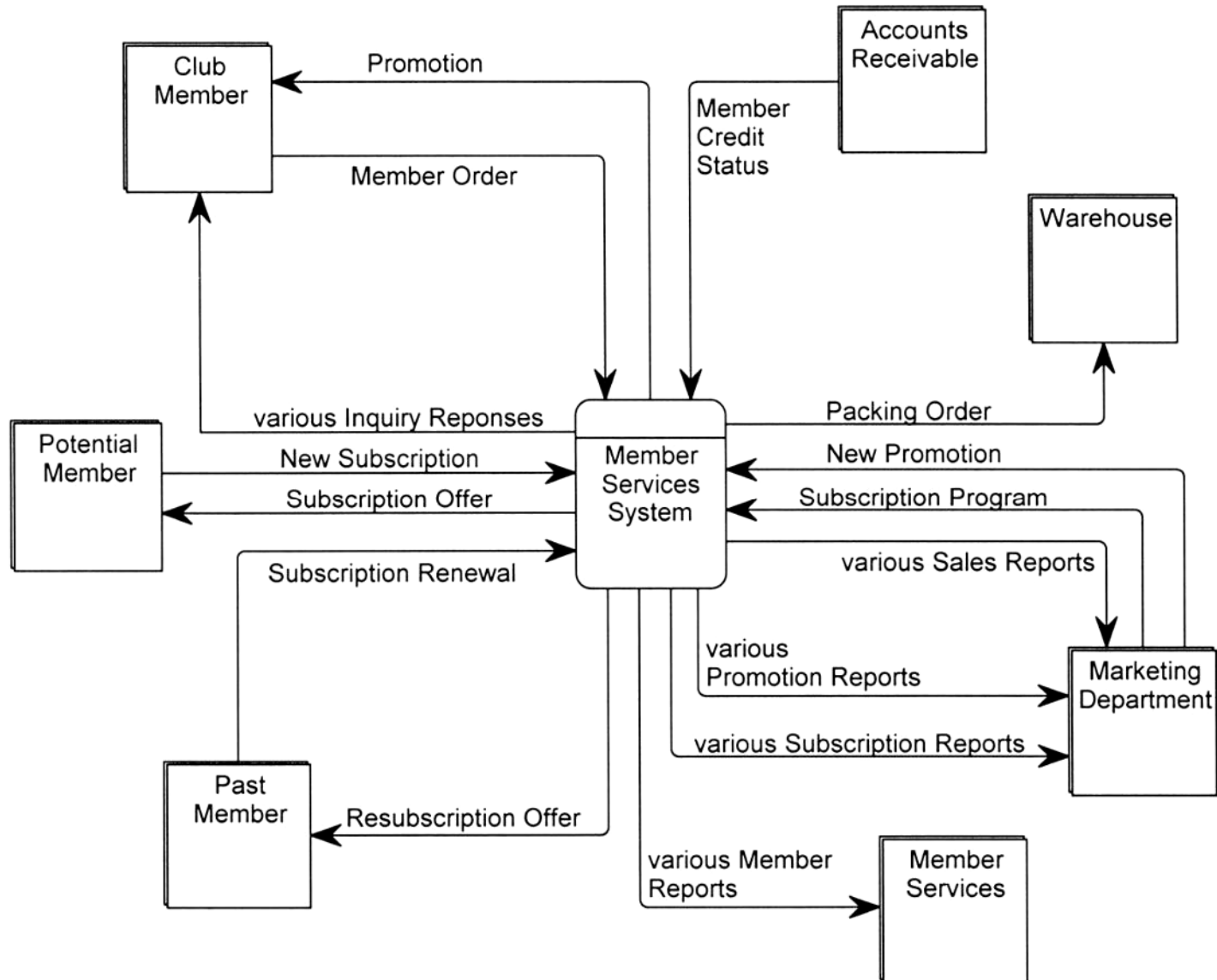


Decomposition

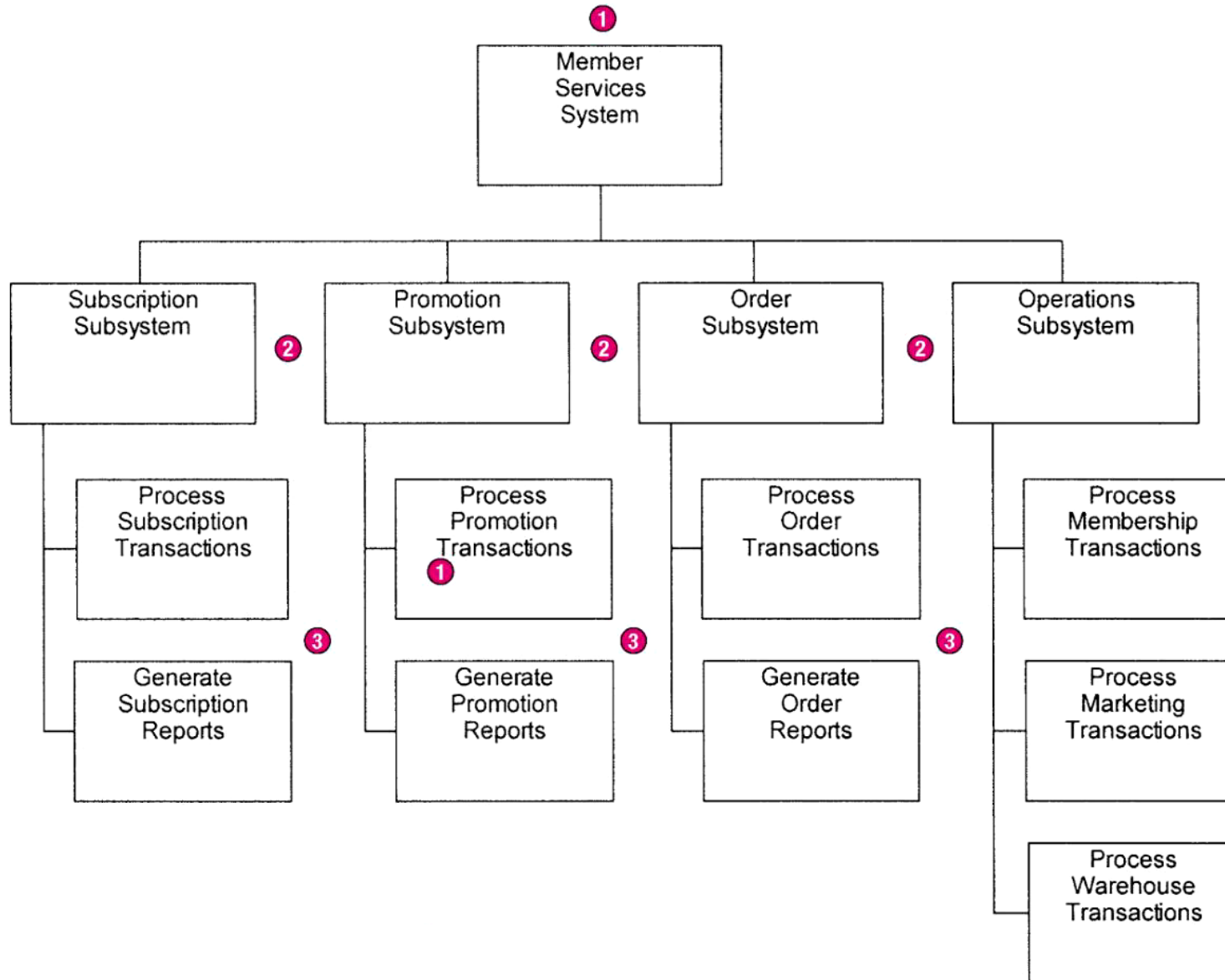
Diagrams



Example : SoundStage Context DFD



SoundStage Functional Decomposition Diagram



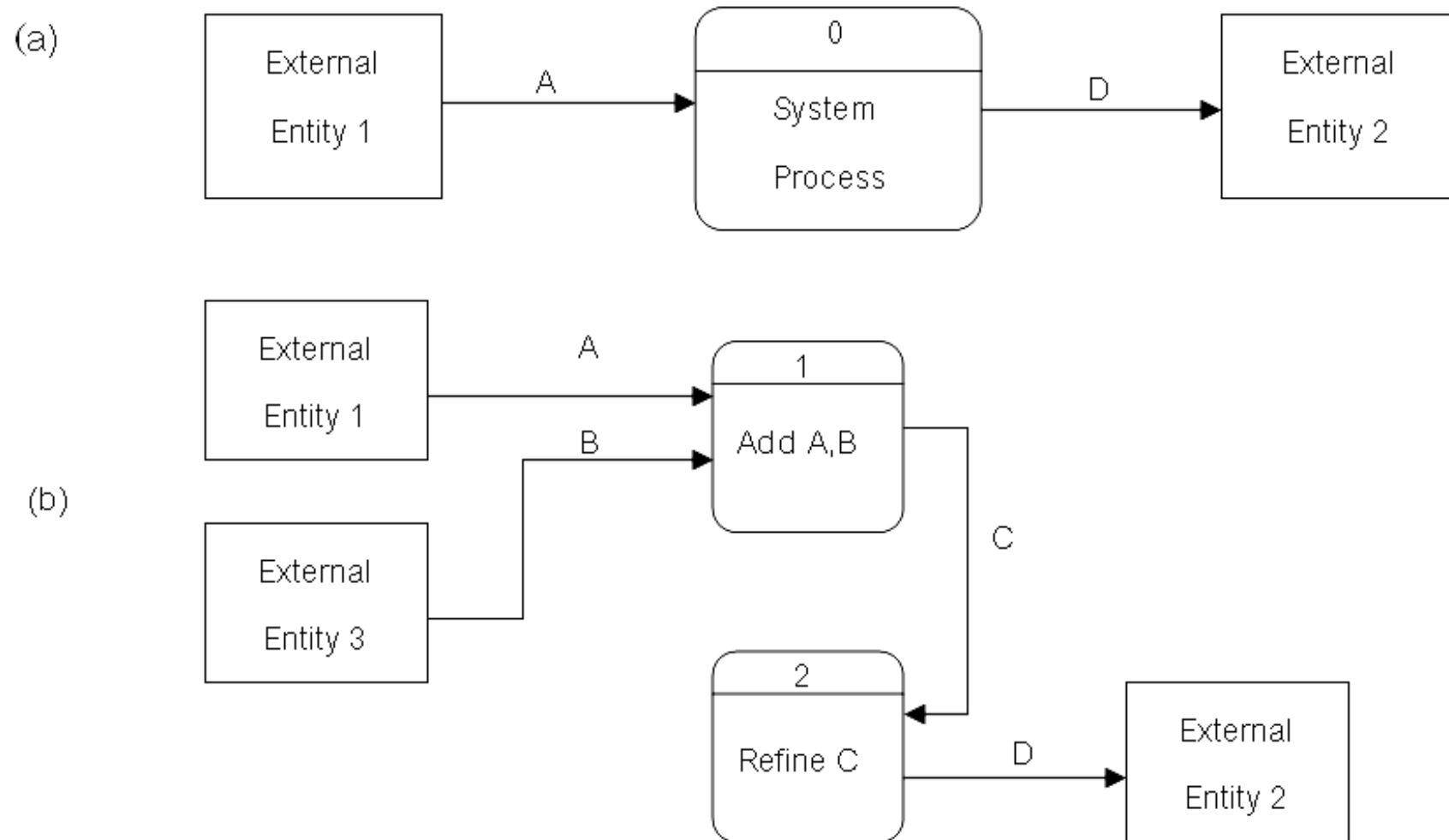
3.5.4 ตรวจสอบสมดุลของ DFD (Balancing DFD)

- เมื่อมีการแบ่งย่อยแผนภาพจากระดับบนลงสู่ระดับล่าง เช่น จาก Level-0 แบ่งย่อยไปใน Level-1 ของ Process 1 SA จะต้องการตรวจสอบความสมดุลของแผนภาพ (Balancing DFD) ด้วย

Balancing DFD (Cont.)

- ❑ Balancing DFD หมายถึง ความสมดุลของแผนภาพกระแสข้อมูลที่จะต้องมีย Input Data Flow ที่เข้าสู่ระบบและ Output Data Flow ที่ออกจากระบบใน DFD ระดับล่างครบทุก Input Data Flow และ Output Data Flow ที่ปรากฏอยู่ใน DFD ระดับบน แต่ในระดับล่างอาจจะมีมากกว่าได้ โดยมีเงื่อนไขว่า Input Data Flow และ Output Data Flow นั้นจะต้องเกิดจาก Process ภายในระดับล่างเท่านั้น และจะนำไปใช้ตรวจสอบความสมดุลของแผนภาพอีกระดับ หากมีการแบ่งย่อยแผนภาพในระดับล่างลงไปอีก

ตัวอย่างการแบ่งย่อยแผนภาพ



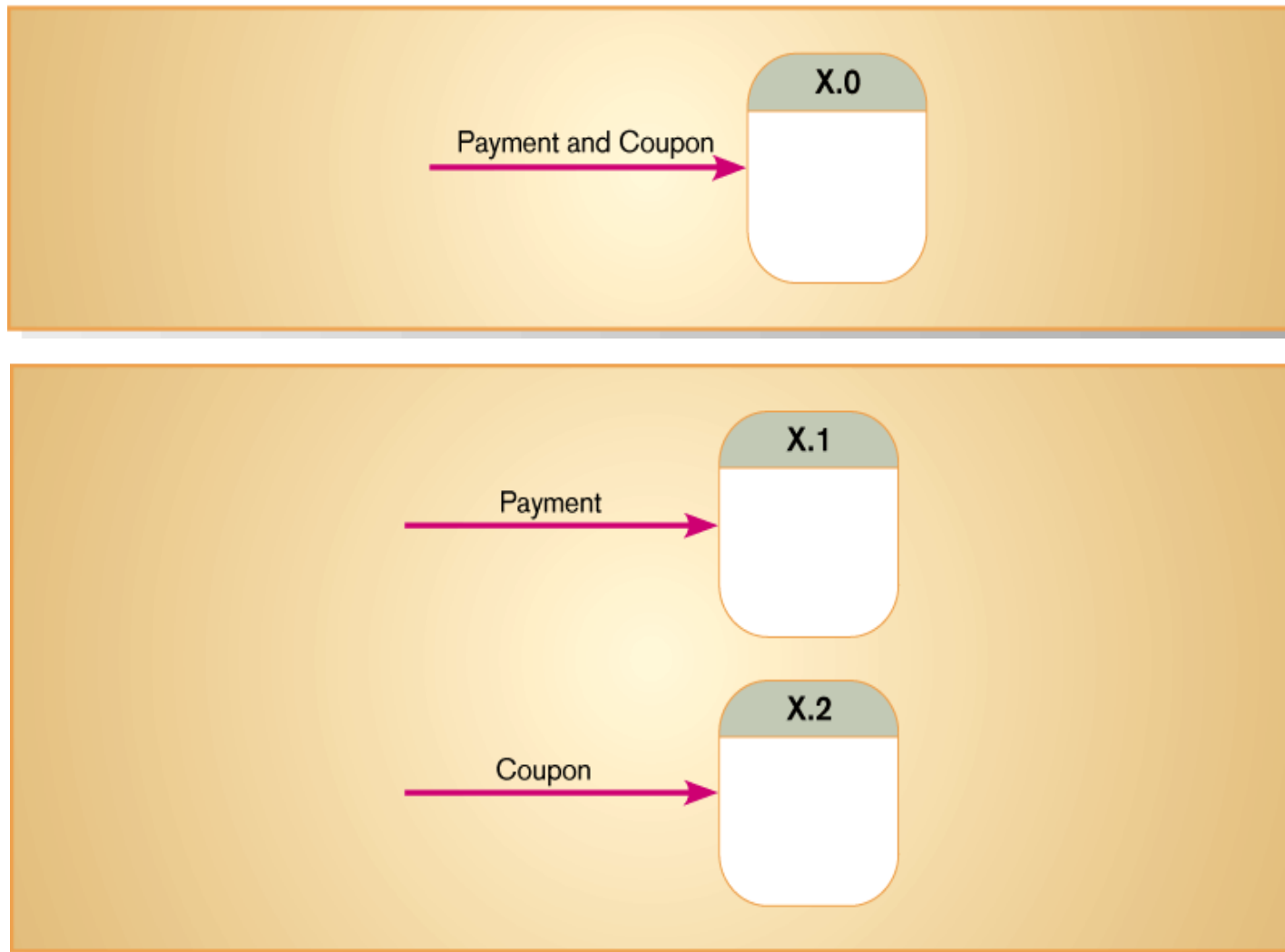
ตัวอย่างการตรวจสอบสมดุลของ DFD

- ❑ จากรูป (a) เป็น Context Diagram ที่มี Input Data Flow เข้าสู่ระบบคือ A จาก External Entity 1 เท่านั้น
- ❑ และมี Output Data Flow คือ D วิ่งไปยัง External Entity 2
- ❑ เมื่อมีการแบ่งย่อยแผนภาพลงที่ Level-0 Diagram ในรูป (b) สังเกตว่ามี Input Data Flow ที่เป็น B จาก External Entity 3 เพิ่มเข้ามา
- ❑ ซึ่งใน Context Diagram ไม่มี External Entity 3

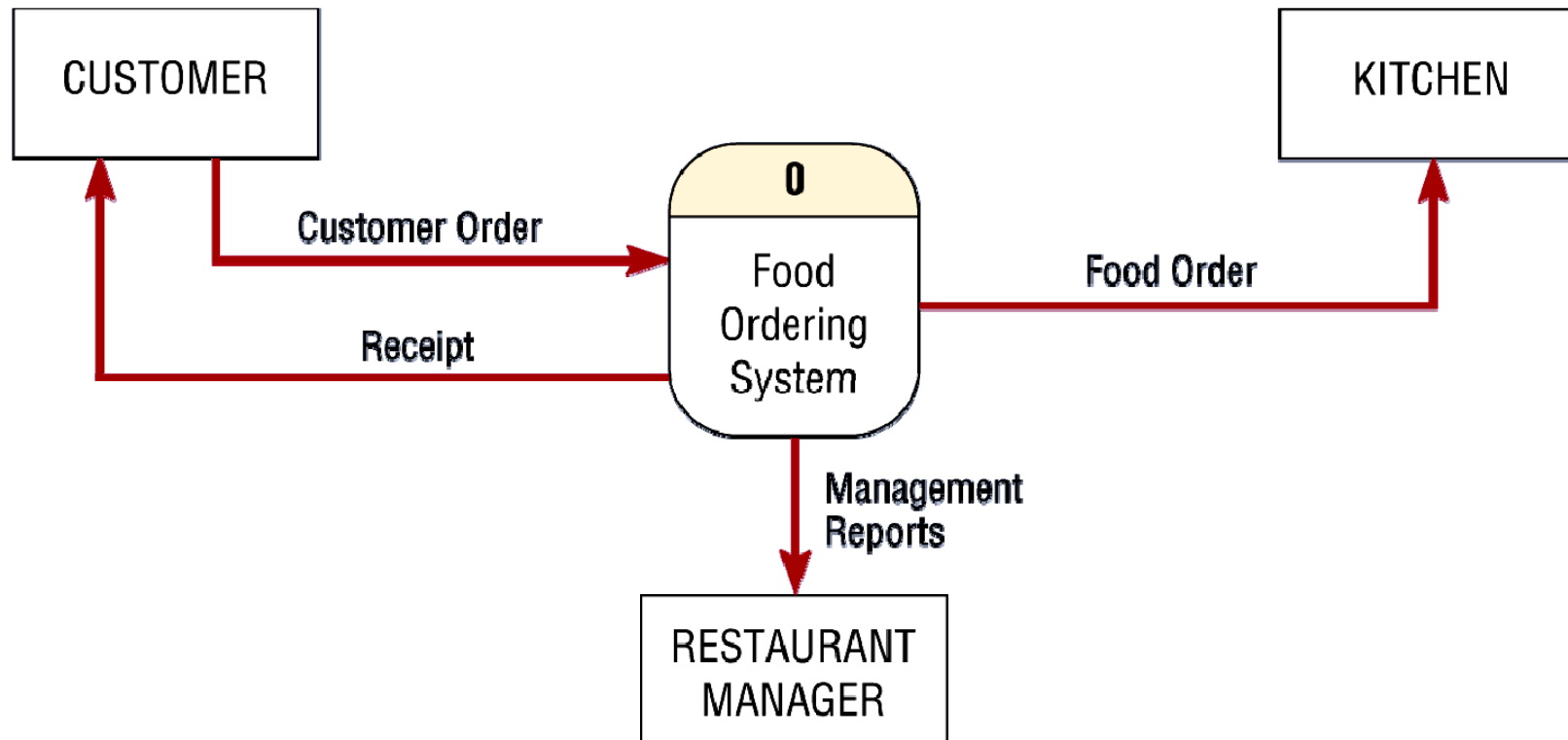
ตัวอย่างการตรวจสอบสมดุลของ DFD (ต่อ)

- ❑ ดังนั้นถือว่า DFD นี้ไม่สมดุล
- ❑ สำหรับ Input Data Flow C สามารถปรากฏอยู่ใน DFD ระดับล่างได้ เนื่องจากเป็น Input Data Flow ที่เกิดจาก Process ภายในระดับล่างนี้เท่านั้น

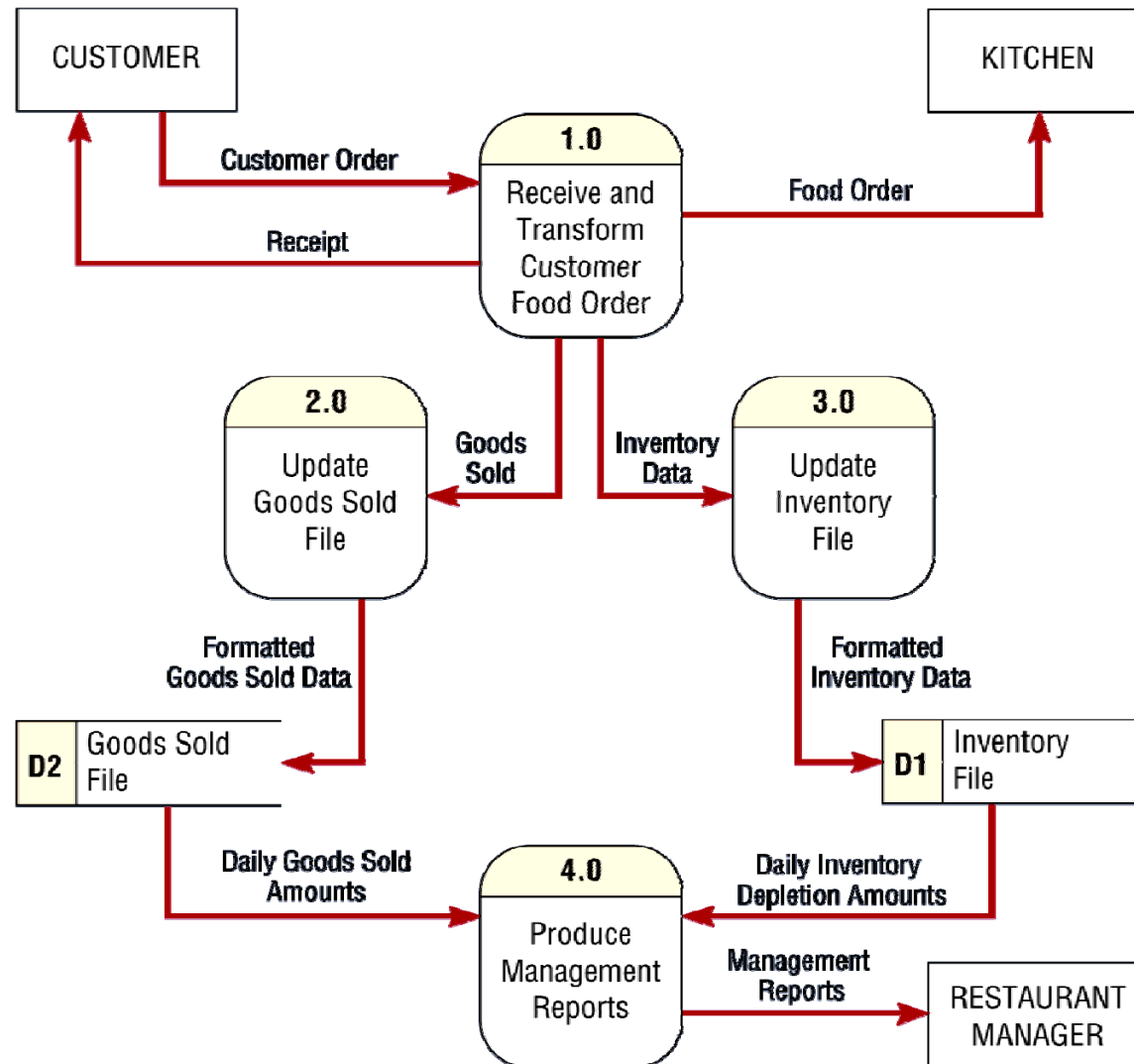
Balancing DFDs



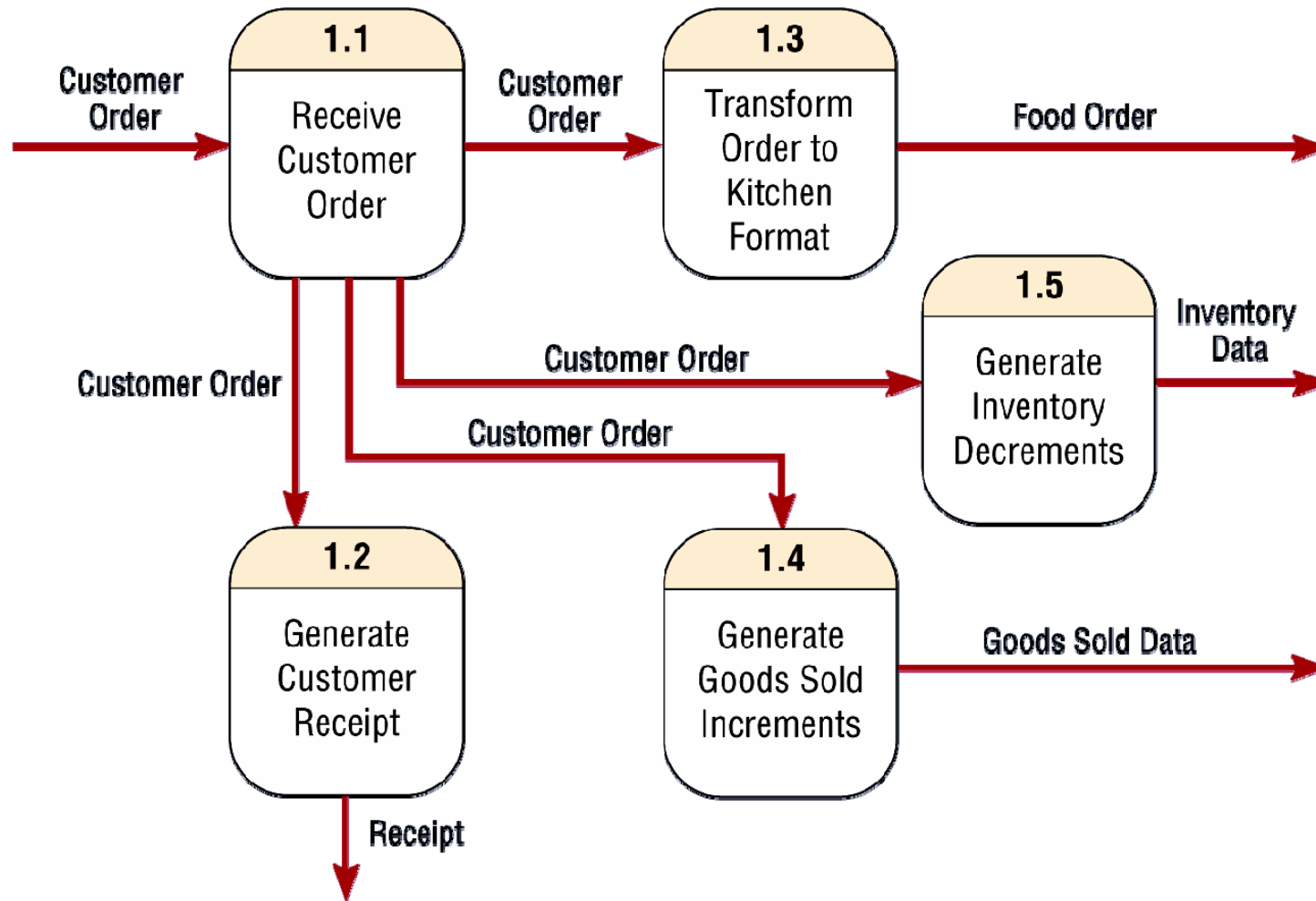
Example : Context Diagram of Food Ordering System



Example : Level-0 DFD of Food Ordering System



Level-1 Diagram Showing Decomposition of Process 1.0 from the Level-0 Diagram



3.6 แนวทางการสร้าง DFD ที่สมบูรณ์

- ❑ เมื่อ SA สร้างแผนภาพกระแสข้อมูลของระบบปัจจุบัน และระบบใหม่ที่จะเสนอให้เป็นทางเลือกในการแก้ไขปัญหาลงแล้ว
- ❑ SA ต้องตรวจสอบ Data Flow, Processes, Data Stores และ External Entity ให้เป็นไปตามกฎและมีการตรวจสอบความสมดุลของแผนภาพ

แนวทางการสร้าง DFD ที่สมบูรณ์ (ต่อ)

- จากนั้น SA ควรมีการตรวจสอบเพิ่มเติมเพื่อให้ได้แผนภาพที่สามารถแสดงให้เห็นถึงรายละเอียดขั้นตอนการทำงาน ข้อมูลที่เกิดจากการประมวลผลแต่ละขั้นตอน และการจัดเก็บข้อมูลได้อย่างถูกต้องและครบถ้วน

หลักเกณฑ์การสร้าง DFD ที่สมบูรณ์

3.6.1 มีความสมบูรณ์ (DFD Completeness)

3.6.2 มีความสอดคล้อง (DFD Consistency)

3.6.3 การทำซ้ำ (Iterative Development)

3.6.4 DFD ระดับล่างสุด (Primitive DFD)

3.6.1 มีความสมบูรณ์ (DFD Completeness)

- ❑ ความหมายของหลักเกณฑ์นี้คือ หากมีการเพิ่มเติมรายละเอียดใดๆ ที่จำเป็นเข้ามาในระบบ SA จะต้องเพิ่มเติมรายละเอียดเหล่านั้นลงใน DFD ด้วยเสมอ
- ❑ หาก Data Flow, Data Store, Process และ External Entity บนแผนภาพ DFD ไม่เชื่อมต่ออยู่กับสิ่งใด ๆ แสดงว่า DFD นั้นไม่สมบูรณ์

3.6.2 มีความสอดคล้อง (DFD Consistency)

- ❑ เป็นความสอดคล้องกันของสิ่งที่ปรากฏอยู่บน DFD ในระดับบนและมีการแบ่งย่อยลงมาในระดับล่าง
- ❑ กล่าวคือ สิ่งที่ปรากฏอยู่บน DFD ในระดับบน เมื่อมีการแบ่งย่อย Process หรือแผนภาพลงมาในระดับล่าง จะต้องมีการปรากฏอยู่ในระดับบนนั้นด้วยเสมอ
- ❑ หลักเกณฑ์นี้จะเกี่ยวข้องกับกฎความสมดุลของแผนภาพ DFD

3.6.3 การทำซ้ำ (Iterative Development)

- ❑ ในการสร้าง DFD ในรอบแรกนั้นจะยังไม่เป็นแผนภาพที่มีความถูกต้องและสมบูรณ์ได้
- ❑ SA จะต้องมีการตรวจสอบแผนภาพหรือมีการปรับปรุงแผนภาพทุกครั้งที่มีการเปลี่ยนแปลงหรือแก้ไขความต้องการ
- ❑ การปรับปรุงแผนภาพนี้จะทำให้มีความถูกต้องมากขึ้นนั่นเอง
- ❑ หากองค์กรใดเลือกใช้ CASE จะทำให้ประหยัดเวลาในส่วนนี้ไป

3.6.4 DFD ระดับล่างสุด (Primitive DFD)

- ❑ เมื่อมีการแบ่งย่อยแผนภาพ DFD ลงมาที่ระดับล่าง เพื่ออธิบายรายละเอียดของขั้นตอนการทำงานภายในระบบ ปัญหาที่เกิดขึ้นคือ “ควรจะสิ้นสุดการแบ่งย่อย Process เมื่อใด”
- ❑ หลักเกณฑ์โดยทั่วไปที่ใช้ในการตัดสินใจว่า เมื่อใดที่ควรจะหยุดแบ่งย่อย Process คือ “เมื่อไม่สามารถแบ่งย่อย Process ได้อีกแล้ว”

หลักเกณฑ์ที่ใช้ในการตัดสินใจเมื่อใดที่ควรหยุดแบ่งย่อย Process

1. เมื่อมีการแบ่งย่อย Process แต่ละ Process ลงมาจนกระทั่งมีการทำงานใน Process นั้นเพียงหน้าที่เดียว เช่น มีการอ่านข้อมูล ปรับปรุง สร้าง และลบข้อมูลในฐานข้อมูล เป็นต้น
2. เมื่อแต่ละ Data Store ที่ใช้ในการจัดเก็บข้อมูล มีการจัดเก็บข้อมูลเพียงไฟล์เดียว เช่น ไฟล์ลูกค้า ไฟล์สินค้า หรือไฟล์สั่งซื้อ เป็นต้น
3. เมื่อผู้ใช้ระบบเห็นว่าไม่มีรายละเอียดใด ๆ ที่จะเป็นต่อการทำงานของระบบแล้ว

3.7 Logical DFD และ Physical DFD

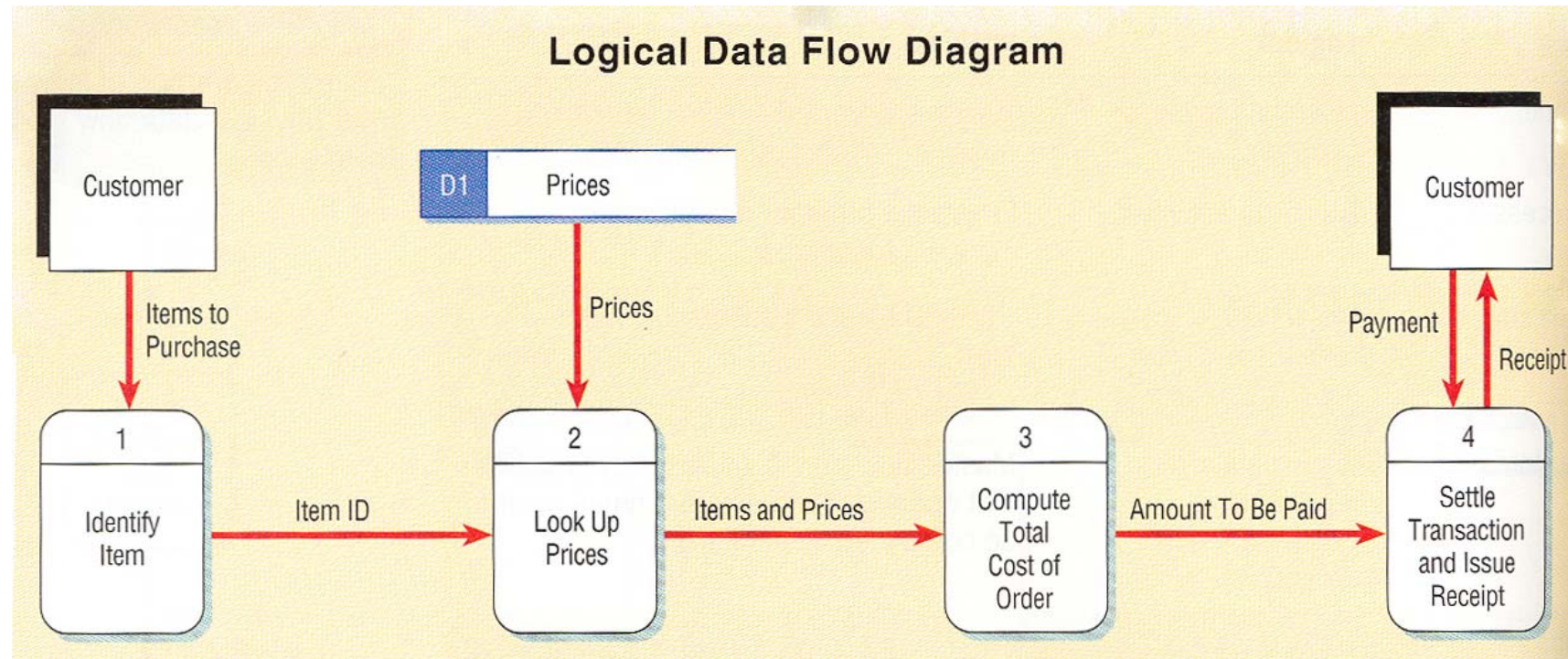
Logical DFD

- ❑ เน้นวิเคราะห์ข้อมูลว่าในระบบหนึ่งมีอะไรเกิดขึ้น
- ❑ มีการเชื่อมโยงกับระบบอื่นอย่างไร
- ❑ ไม่แสดงให้เห็นอุปกรณ์ สื่อจัดเก็บ แฟ้ม และการควบคุมความปลอดภัย

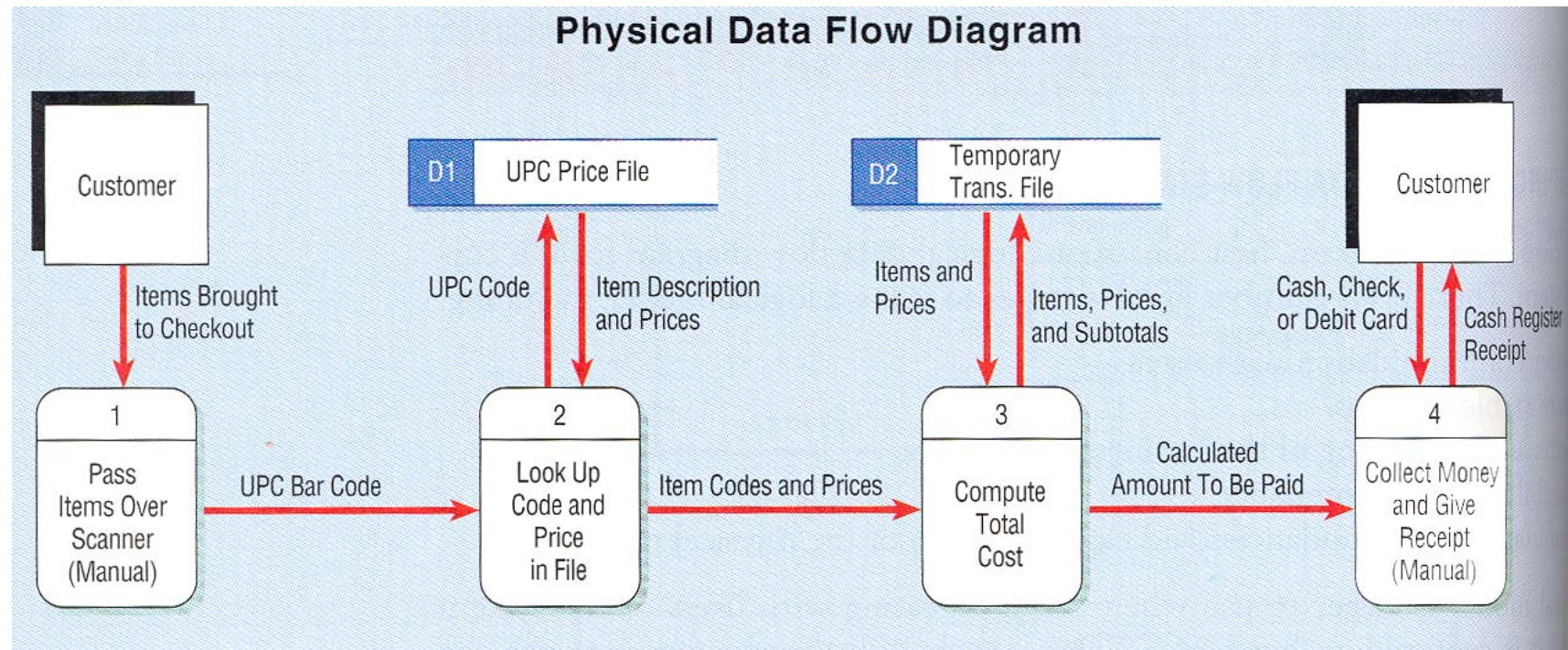
Physical DFD

- ❑ วิเคราะห์เพิ่มเติมจากส่วน Logical
- ❑ แสดงให้ทราบถึงรายละเอียด Master File และ Transaction File

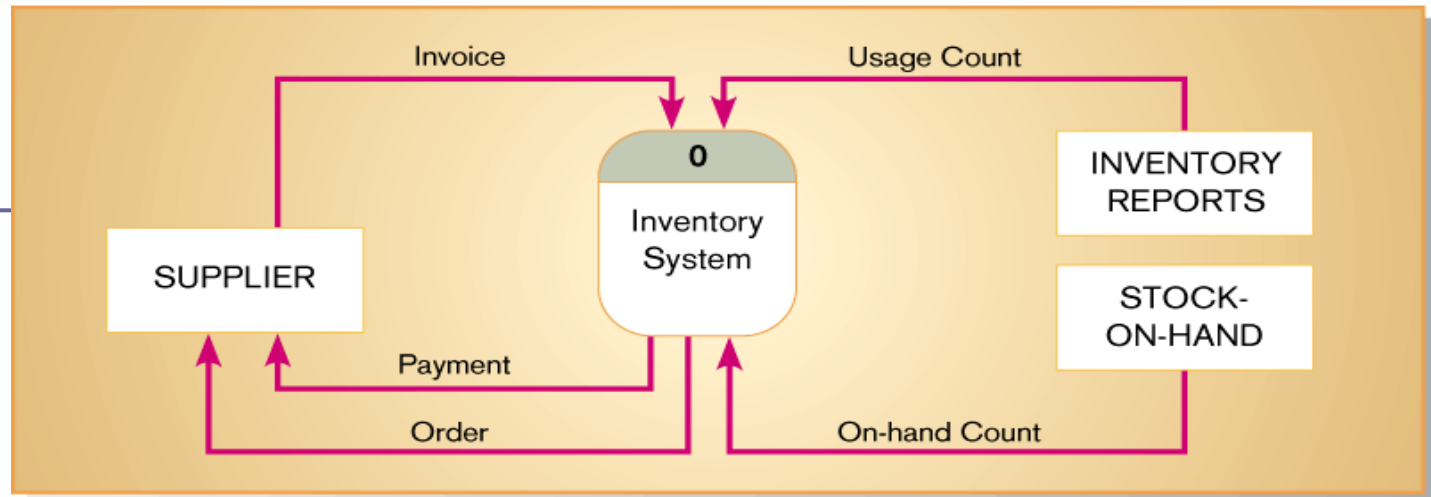
ตัวอย่าง Logical DFD



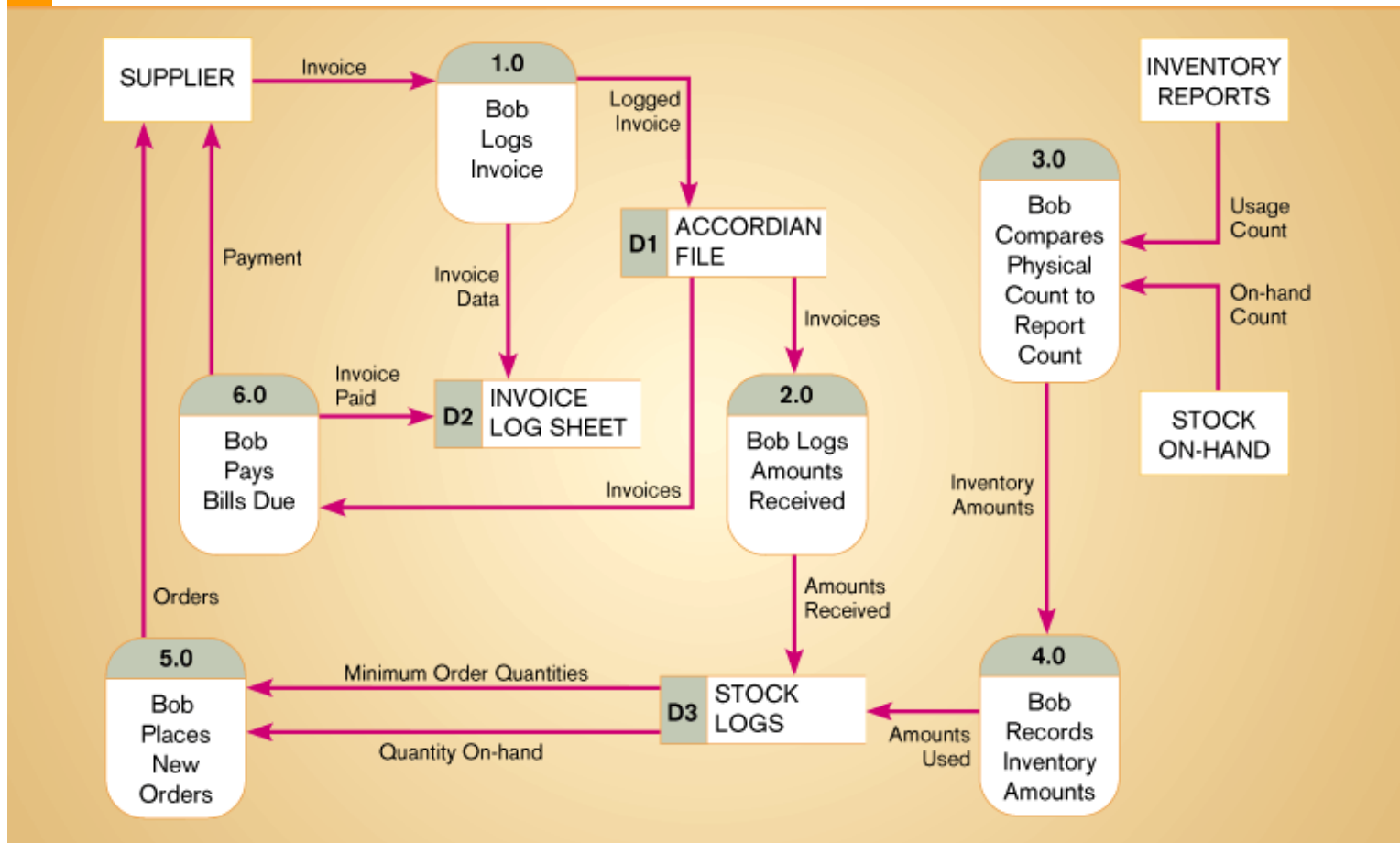
ตัวอย่าง Physical DFD



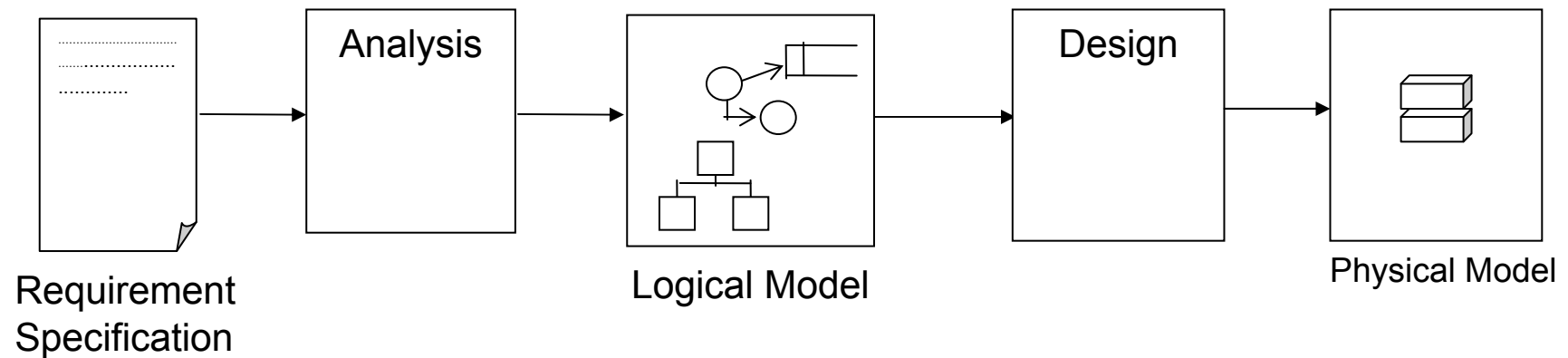
Current Physical System



Level 0 Physical System



ขั้นตอนการวิเคราะห์เพื่อไปสู่การออกแบบ



3.8 คำอธิบายการทำงานภายในโปรเซส

- ❑ Process Description หรือ Process Specification หรือ Mini-Specs
- ❑ ใช้แสดงรายละเอียดภายในของแต่ละโปรเซสว่าโปรเซสต่าง ๆ เหล่านี้มีกระบวนการทำงานอย่างไร (Process Logic) โดยที่รายละเอียดเหล่านี้ไม่สามารถแสดงให้เห็นภายใน DFD ได้

วัตถุประสงค์ของคำอธิบายการประมวลผล

1. เพื่อลดความไม่ชัดเจนหรือความกำกวมของโพรเซส
จุดประสงค์ที่สำคัญคือ ใช้เป็นกฎเกณฑ์หรือข้อบังคับใช้ของ
นักวิเคราะห์ เพื่อการเรียนรู้รายละเอียดของแต่ละโพรเซส
ว่ามีกระบวนการทำงานอย่างไรเป็นสำคัญ
2. เพื่อความเที่ยงตรงและสามารถทำให้สำเร็จตามกระบวนการ
ที่อธิบายไว้อย่างถูกต้องและเข้าใจตรงกัน
3. เพื่อใช้สำหรับการตรวจสอบในขั้นตอนของการออกแบบ
ระบบต่อไป

ระดับของคำอธิบายการประมวลผล

1. ระดับการใช้งาน (Usage Level)

- ❑ มุ่งเน้นรายละเอียดด้านการปฏิบัติงานของผู้ใช้เป็นสำคัญ
- ❑ มักอยู่ในรูปแบบของถ้อยคำ เรียงความ เพื่อใช้อธิบายการประมวลผล หรือที่เรียกว่าภาษาธรรมชาติ

ตัวอย่างการเขียนคำอธิบายการประมวลผลในรูปแบบภาษาธรรมชาติ

Process Description	

ระดับของคำอธิบายการประมวลผล (ต่อ)

2. ระดับระบบ (System Level)

- ❑ มุ่งเน้นความเที่ยงตรงในรายละเอียด
- ❑ เพื่อเป็นไปตามข้อกำหนดของระบบ (System Specification)
- ❑ ง่ายต่อการนำไปแปลงเป็นภาษาคอมพิวเตอร์
- ❑ เช่น คำอธิบายการประมวลผลในรูปแบบภาษาสคริปต์ (ภาษาอังกฤษแบบโครงสร้าง, Decision Tree, Decision Table)
- ❑ ทำให้โปรแกรมเมอร์สามารถนำไปประยุกต์ใช้กับการเขียนโปรแกรมได้ทันที

ภาษาอังกฤษแบบมีโครงสร้าง (Structured English)

- ❑ เป็นการนำภาษาอังกฤษมาเขียนเพื่อบ่งบอกรายละเอียดการทำงานของโปรเซสที่ปรากฏอยู่ใน DFD
- ❑ รูปแบบการเขียนใกล้เคียงกับไวยากรณ์ที่ใช้ในการเขียนโปรแกรม

ภาษาอังกฤษแบบโครงสร้าง

- ประโยคอาจอยู่ในรูป Algebraic Equation

$$X = (Y*Z)/(Q+14)$$

- ประโยคประกอบด้วยคำกริยาและคำนาม (แสดงถึง Object) คำกริยาควรจำกัด เช่น

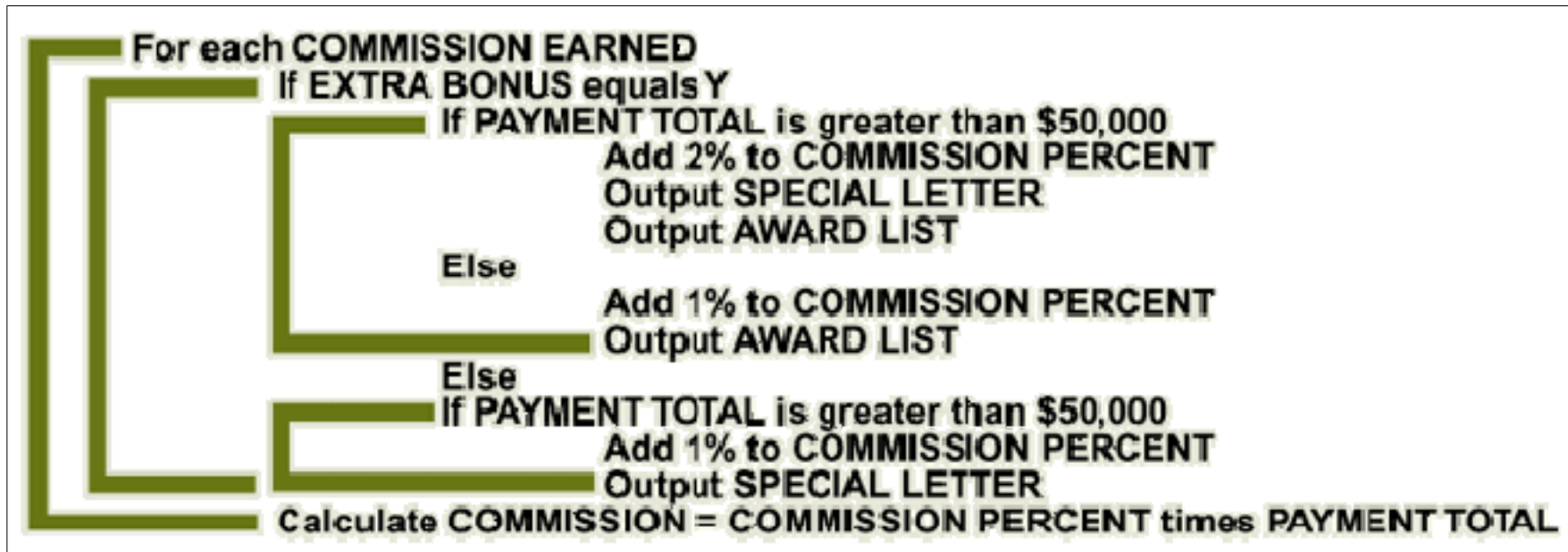
- GET (or ACCEPT or READ)
- PUT (or DISPLAY or WRITE),
- FIND (or SEARCH or LOCATE)
- ADD, SUBTRACT, MULTIPLY, DIVIDE
- COMPUTE, DELETE, VALIDATE,
- MOVE, REPLACE, SET, SORT

ภาษาอังกฤษแบบโครงสร้าง (ต่อ)

□ ประโยคประกอบร่วมกันโดยใช้หลักของภาษาโครงสร้าง

- Sequence
- Selection
 - ↳ IF...THEN...ELSE
 - ↳ CASE
- Iteration
 - ↳ WHILE...DO
 - ↳ FOR...DO

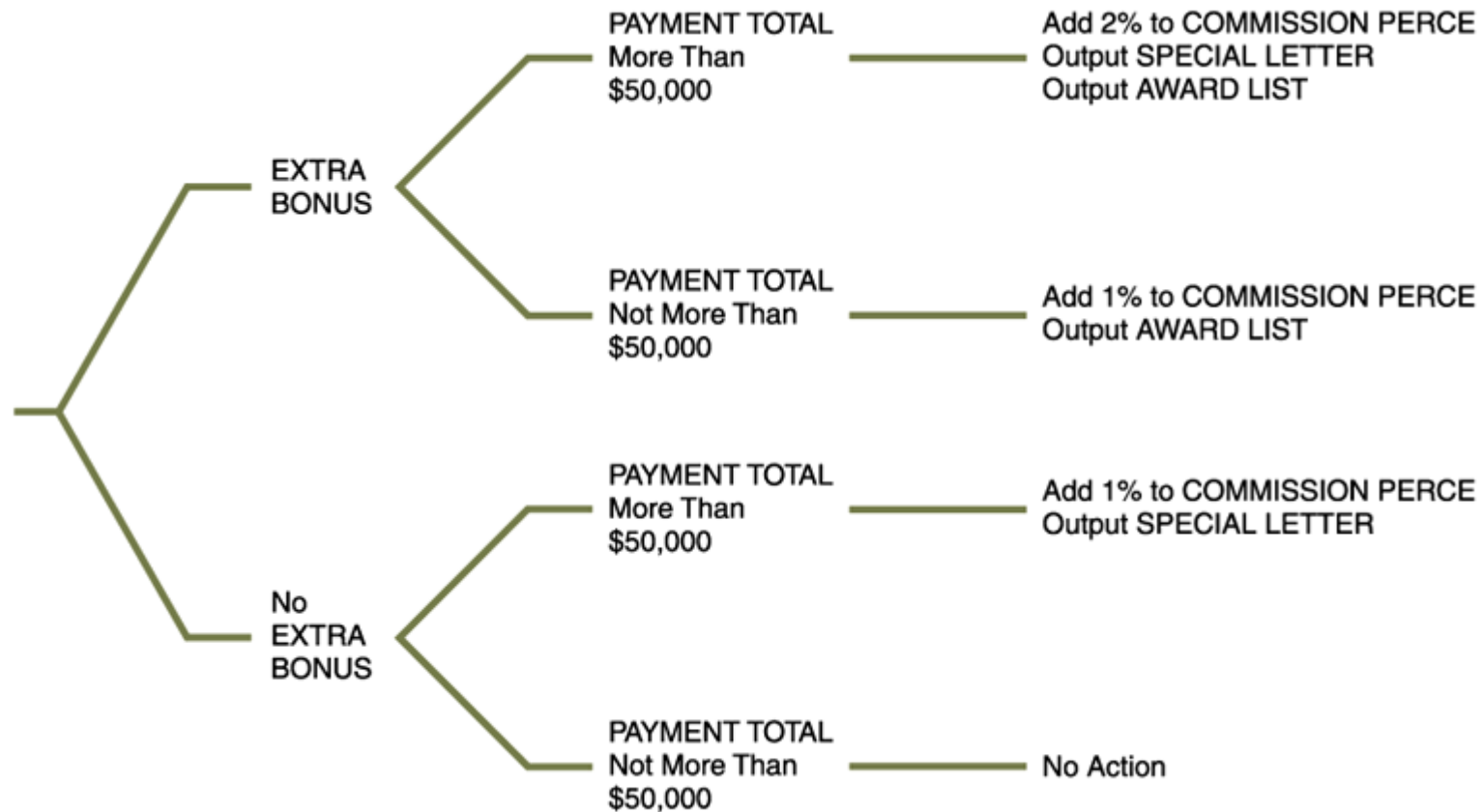
ตัวอย่างการเขียนคำอธิบายการประมวลผล ในรูปแบบภาษาอังกฤษแบบโครงสร้าง



ต้นไม้การตัดสินใจ (Decision Tree)

- ❑ เป็นเครื่องมือช่วยในการตัดสินใจในลักษณะของต้นไม้
- ❑ แสดงเงื่อนไขต่าง ๆ เมื่อเงื่อนไขเป็นจริงจะต้องทำอะไรต่อไปตามขั้นตอน ถ้าไม่เป็นจริงจะต้องทำอะไร

ตัวอย่างการเขียนคำอธิบายการประมวลผล ในรูปแบบ Decision Tree



ตัวอย่างที่ 2 – ต้นไม้การตัดสินใจ

- ❑ ในการรับสมัครคนเข้าทำงานมีกฎเกณฑ์ดังนี้
 - 1) เป็นเพศชาย
 - 2) อายุมากกว่า 25 ปี
 - 3) ผ่านการเกณฑ์ทหารแล้ว
 - 4) วุฒิปริญญาตรีขึ้นไป
 - 5) จบสาขาคอมพิวเตอร์ได้รับเงินเดือน 30,000 บาท
 - 6) จบสาขาอื่นได้รับเงินเดือน 20,000 บาท

ตารางการตัดสินใจ (Decision Table)

- ❑ มีลักษณะเป็นตาราง 2 มิติ
- ❑ แสดงเงื่อนไขการกระทำและความเป็นไปได้ต่าง ๆ ที่จะเกิดขึ้น โดยจะตรวจสอบเงื่อนไขแต่ละเงื่อนไข ซึ่งจะช่วยในการวิเคราะห์และอธิบายการทำงานได้ชัดเจนขึ้น

ตัวอย่างการเขียนคำอธิบายการประมวลผล ในรูปแบบ Decision Table

VERIFY ORDER Process					
1 → Based On	1	2	3	4	
2 → Credit status is OK	Y	Y	N	N	3 ← }
Product is in stock	Y	N	Y	N	
Accept order	X				4 ← }
Reject order		X	X	X	

ตัวอย่างที่ 3 - ตารางการตัดสินใจ

- ❑ การพิจารณาให้เงินกู้ของธนาคารแห่งหนึ่ง มีการพิจารณาเงื่อนไขต่าง ๆ ดังนี้
 - 1) ประเภทธุรกิจ (อาหาร ไฟฟ้า สิ่งทอ)
 - 2) ระยะเวลาที่ติดต่อกับธนาคาร (<1 ปี, ≥ 1 ปี)
 - 3) การส่งเสริมการลงทุน (ได้, ไม่)
 - 4) ธุรกิจอาหาร และไฟฟ้าที่ได้รับการส่งเสริมการลงทุน และติดต่อกับธนาคารนานกว่า 1 ปี ให้วงเงินกู้เกิน 10 ล้าน ถ้าไม่ถึง 1 ปีให้วงเงินกู้ไม่เกิน 10 ล้าน
 - 5) ธุรกิจสิ่งทอที่ได้รับการส่งเสริมการลงทุนให้วงเงินกู้ไม่เกิน 10 ล้าน

3.9 พจนานุกรมข้อมูล (Data Dictionary)

- เป็นสิ่งที่บอกคุณลักษณะของข้อมูลที่ใช้ในองค์กร
- เป็นสิ่งสำคัญสำหรับ SA ในการพัฒนาระบบ เนื่องจากช่วยให้ทีมงานและผู้ใช้ระบบทุกคนเข้าใจตรงกัน


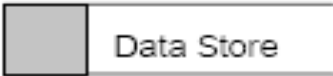
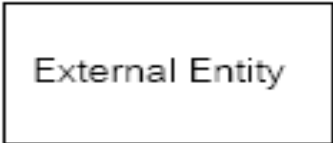

สิ่งที่ควรพิจารณา

1. การเขียนอธิบาย Data Flow และ Data Store ควรจะอธิบายถึงโครงสร้างข้อมูลและสมาชิกของข้อมูล (Data Structure and Data Elements)
2. การเขียนอธิบาย Process ควรอธิบายถึง Logic ของ Process โดยเขียนเป็น Process Specification และ Structured decisions
3. การเขียนอธิบาย Entity ควรอธิบายถึงความสัมพันธ์ระหว่าง Entity ให้ชัดเจนโดยใช้ ERD ช่วย

ข้อควรระวัง

1. ไม่ควรอธิบายซ้ำซ้อนกัน (Redundancy) เช่น สิ่งใดที่มีใน Process Specification แล้ว ก็ไม่ควรนำมาเขียนใน DD
2. ถ้าใน DFD มีรายละเอียดอยู่แล้วก็ไม่จำเป็นต้องเขียนลงใน DD เช่นกัน
3. ควรตรวจสอบความสอดคล้องกัน (Consistency) ระหว่าง DD และ DFD ด้วย

แสดงความสัมพันธ์ระหว่าง DFD และ DD

Data Flow Diagram	Data Dictionary
 <p>Data Flow</p>	<div> <div>Data Flow Description xxx</div> <div>Data Structure xxx</div> <div>Data Elements xxx</div> </div>
 <p>Data Store</p>	<div> <div>Data Store Description xxx</div> <div>Data Structure xxx</div> <div>Data Elements xxx</div> </div>
 <p>External Entity</p>	<div>External Entity Description</div>
 <p>Process</p>	<div>Process Description</div>

ความสำคัญของพจนานุกรมข้อมูล

1. สำหรับระบบที่ใหญ่จะมีปริมาณข้อมูลที่ไหลไปมาระหว่างระบบงานมาก ทุกระบบทำงานตลอดเวลา ผู้ใช้ไม่สามารถจดจำรายละเอียดเกี่ยวกับระบบได้ทั้งหมด ดังนั้นพจนานุกรมจะเป็นสิ่งที่ช่วยบันทึกรายละเอียดต่าง ๆ
2. ช่วยให้ผู้ใช้ระบบทุกคน เข้าใจสมาชิกข้อมูล และกิจกรรมของระบบในความหมายเดียวกัน

ความสำคัญของพจนานุกรมข้อมูล (ต่อ)

3. ใช้เป็นคู่มืออำนวยความสะดวก ในการวิเคราะห์ลักษณะต่าง ๆ ของระบบดังนี้

- ☐ ระบบ Transaction
- ☐ การโต้ตอบเกี่ยวกับสารสนเทศต่าง ๆ (Inquiries)
- ☐ การออกแบบรายงานในรูปแบบที่ผู้ใช้ยอมรับ
- ☐ การออกแบบแฟ้มข้อมูล
- ☐ การกำหนดความสามารถของระบบว่ารับข้อมูลหรือประมวลผลได้มากน้อยเท่าใด

ประเภทของพจนานุกรมข้อมูล

3.9.1 พจนานุกรมกระแสการไหลของข้อมูล

(Data Flow Dictionary)

3.9.2 พจนานุกรมโครงสร้างข้อมูล

(Data Structure Dictionary)

3.9.3 พจนานุกรมอธิบายสมาชิกข้อมูล

(Data Elements Dictionary)

3.9.4 พจนานุกรมแหล่งเก็บข้อมูล

(Data Store Dictionary)

3.9.1 พจนานุกรมกระแสการไหลของข้อมูล

- Data Flow มักเป็นสิ่งที่ถูกนำมาเขียน DD เพราะเป็นตัวกำหนด Input/Output จากการวิเคราะห์ระบบ

พจนานุกรมกระแสการไหลของข้อมูลประกอบด้วย

1. ชื่อข้อมูล (Data names)
2. คำอธิบายย่อ ๆ เกี่ยวกับข้อมูลนั้น (Data Descriptions)
3. ชื่ออีกชื่อหนึ่ง (Aliases)
4. แหล่งของ Data Flow
5. ชื่อของ Data Structure และ Data Elements ที่ใช้ใน Data Flow นั้น ๆ
6. ปริมาณของข้อมูลในช่วงเวลาที่สนใจ

ตัวอย่างพจนานุกรมกระแสการไหลของข้อมูล

Data Flow Description Form (แบบฟอร์มรายละเอียดของเส้นกระแสข้อมูล)

Name ใบสั่งซื้อสินค้าของลูกค้า

Description มีรายละเอียดเกี่ยวกับข้อมูลการสั่งซื้อของลูกค้า ซึ่งข้อมูลเหล่านี้จะนำมาใช้เพื่อปรับปรุงข้อมูลเพิ่มข้อมูลลูกค้าและเพิ่มข้อมูลสินค้า และนำไปสร้างเรคคอร์ดการสั่งซื้อ โดยใบสั่งซื้อหนึ่งใบจะเป็นของลูกค้าเพียงคนเดียว โดยใบสั่งซื้อนี้อาจได้จากการส่งเมลล์, โทรสาร (Fax), หรือลูกค้าทำการโทรศัพท์ไปที่แผนกที่รับการสั่งซื้อโดยตรง

Alias รายการสั่งซื้อสินค้า

Source ลูกค้า Destination รับข้อมูลการสั่งซื้อ

Type of Data Flow

☐ File

☒ Screen

☐ Report

☐ Form

☐ Internal

Data Structure Traveling with the Flow ข้อมูลการสั่งซื้อ

Volume and Frequency/ Time 10 ครั้ง/ ชั่วโมง

3.9.2 พจนานุกรมโครงสร้างข้อมูล

- เป็นการเขียนคำอธิบายหรือรายละเอียดของข้อมูลโครงสร้าง (Data Structure) ว่าประกอบไปด้วยข้อมูลย่อยหรือข้อมูลเดี่ยว (Data Element) อะไรบ้าง เพื่อความเข้าใจในระบบงานให้ชัดเจนมากขึ้น

สัญลักษณ์ที่ใช้ในพจนานุกรมโครงสร้างข้อมูล

เครื่องหมาย	ความหมาย
=	ประกอบด้วย หรือเท่ากับ (Definition)
+	และ
()	ข้อความในวงเล็บถือเป็น Option จะมีหรือไม่มีก็ได้
{ }	ให้กระทำตามข้อมูลในวงเล็บซ้ำ (Iteration)
[]	ให้เลือกข้อมูลพื้นฐาน (Data Elements) ตัวใดตัวหนึ่งในวงเล็บ
...	ข้อความต่อไปนี้คือคำอธิบาย (Comment)
@	ข้อมูลต่อไปนี้คือ key field
/, :	ใช้เป็นเครื่องหมายแยกข้อมูลให้เลือกได้ใน []

ตัวอย่างใบสั่งซื้อสินค้าของลูกค้า

บริษัท เวิลด์เทรนด์ จำกัด (World's Trend)
24 ถ.สุขเกษม ซอย 10 อ.เมือง จ.สกลนคร

ใบสั่งซื้อสินค้าของลูกค้า

คำนำหน้าชื่อ : นางสาว ชื่อ : พณรัตน์ นามสกุล : ศรีเชษฐา

บ้านเลขที่ : 57/357 ถนน : นิมิตรใหม่ ตำบล : มีนบุรี อำเภอ : จังหวัด : กรุงเทพฯ

รหัสไปรษณีย์ : 10510 โทรศัพท์ : 2548415

หมายเลขลูกค้า : 09288 หมายเลขแคตตาล็อก : 9401A วันที่สั่งซื้อ (MM/DD/YY) : 05/25/99

รหัสสินค้า	รายละเอียดสินค้า	ปริมาณ	ขนาด	สี	ราคา	รวม
12343	สูท Jogging	1	M	BL	350.50	350.50
54224	ถุงเท้า Cushion impact	4	M	WH	40.25	170.00
10617	กางเกง Running	1	M	BL	120.25	120.25
10617	กางเกง Running	1	M	GR	120.25	120.25
วิธีการจ่ายเงิน <input type="checkbox"/> เช็ค <input checked="" type="checkbox"/> บัตรเครดิต <input type="checkbox"/> เงินสด					ราคารวม	770.00
ให้เลือกข้อต่อไปนีถ้านำมาจ่ายโดยบัตรเครดิต					ภาษี	0.00
<input checked="" type="checkbox"/> World's Trend <input type="checkbox"/> AmExpress <input type="checkbox"/> Visa <input type="checkbox"/> Discover					ค่าขนส่ง	90.50
หมายเลขบัตรเครดิต วันที่หมดอายุบัตร					รวมตามใบสั่ง	860.50

ตัวอย่างพจนานุกรมโครงสร้างข้อมูล

Data Structure Description Form

(แบบฟอร์มส่วนอธิบายโครงสร้างข้อมูล)

ใบสั่งซื้อของลูกค้า = หมายเลขลูกค้า + ชื่อลูกค้า + ที่อยู่ลูกค้า + หมายเลขแคตตาล็อก + วันที่สั่งซื้อ
+ {รายการสั่งซื้อ} + ราคารวม + (ภาษี) + ค่าขนส่ง + รวมตามใบสั่ง
+ วิธีการจ่ายเงิน + (ชนิดของบัตรเครดิต) + (หมายเลขบัตรเครดิต) + (วันที่หมดอายุบัตร)

ชื่อลูกค้า = คำนำหน้าชื่อ + ชื่อ + นามสกุล

ที่อยู่ลูกค้า = บ้านเลขที่ + ถนน + ตำบล + อำเภอ + จังหวัด + รหัสไปรษณีย์ + โทรศัพท์

รายการสั่งซื้อ = รหัสสินค้า + รายละเอียดสินค้า + ปริมาณ + ขนาด + สี + ราคา + รวม

วิธีการจ่ายเงิน = [เช็ค | บัตรเครดิต | เงินสด]

ชนิดของบัตรเครดิต = [World's Trend | AmExpress | Visa | Discover]

3.9.3 พจนานุกรมอธิบายสมาชิกข้อมูล

□ เป็นการอธิบายรายละเอียดของสมาชิกของข้อมูลในโครงสร้างข้อมูล ประกอบด้วย

1. หมายเลขข้อมูล
2. ชื่อข้อมูล ควรจะเป็นชื่อที่ไม่ซ้ำกัน ซึ่งจะเป็นชื่อที่ใช้ในการผลิตโปรแกรม
3. ชื่อเล่นหรือชื่ออื่น ๆ ของข้อมูล (Alias)
4. คำอธิบายเกี่ยวกับข้อมูลนั้น ๆ
5. คุณลักษณะของข้อมูล ซึ่งได้แก่ ความยาวของข้อมูลนั้น ๆ รูปแบบของข้อมูล ชนิดของข้อมูล
6. ขอบเขตหรือค่าที่เป็นไปได้ของข้อมูล

ตัวอย่างพจนานุกรมแสดงรายละเอียดสมาชิกของข้อมูล

Data Element Description Form (แบบฟอร์มรายละเอียดสมาชิกของข้อมูล)		
ID _____		
Name <u>หมายเลขลูกค้า</u>		
Description <u>รหัสลูกค้าจะเป็นตัวบ่งชี้ลูกค้าแต่ละคน ซึ่งลูกค้าแต่ละคนจะไม่มีรหัสลูกค้าซ้ำกับของคนอื่น โดยรหัสนี้</u> <u>จะถูกสร้างขึ้นจากรายการเปลี่ยนแปลงของธุรกิจภายใน 5 ปี</u>		
<hr/>		
Length <u>6</u>	Dec.Pt. _____	<input type="checkbox"/> Alphabetic
Input Format <u>9(6)</u>		<input type="checkbox"/> Alphanumeric
Output Format <u>9(6)</u>		<input type="checkbox"/> Date
Default Value <u>-</u>		<input checked="" type="checkbox"/> Numeric
<input checked="" type="checkbox"/> Continuous or <input type="checkbox"/> Discrete		<input type="checkbox"/> Base or <input checked="" type="checkbox"/> Derived
<hr/>		
<i>Continuous</i>	<i>Discrete</i>	
	Value	Meaning
Upper Limit <u>< 999999</u>	_____	_____
Lower Limit <u>≥ 0</u>	_____	_____

3.9.4 พจนานุกรมแหล่งเก็บข้อมูล

❑ ข้อมูลพื้นฐานของทุกตัวจะถูกจัดเก็บลงใน Data Store การเขียนอธิบายในพจนานุกรมข้อมูล ควรจะประกอบด้วย

1. ชื่อ Data Store ซึ่งต้องมีชื่อเดียว ไม่ซ้ำกัน
2. ชื่ออีกชื่อหนึ่ง (Alias)
3. คำอธิบาย
4. ชนิดของแฟ้มข้อมูลว่าเป็น Manual หรือใช้คอมพิวเตอร์
5. ถ้าชนิดของแฟ้มข้อมูลเป็นคอมพิวเตอร์ ต้องอธิบายถึงความสัมพันธ์ระหว่างแฟ้มข้อมูล เพื่อนำไปสู่ฐานข้อมูล เช่น คีย์หลัก คีย์รองที่ใช้
6. จำนวนเรคอร์ดที่สูงสุดของแฟ้มข้อมูล อัตราการเจริญเติบโตต่อปี
7. ชื่อ Data Structures

ตัวอย่างพจนานุกรมแหล่งเก็บข้อมูล

Data Store Description Form	
(แบบฟอร์มรายละเอียดของส่วนเก็บข้อมูล)	
ID	D1
Name	แฟ้มข้อมูลหลักลูกค้า
Description	ใช้เก็บเรคคอร์ดลูกค้าแต่ละคน
Data Store Characteristics	
File Type	<input checked="" type="checkbox"/> Computer <input type="checkbox"/> Manual
File Format	<input checked="" type="checkbox"/> Database <input type="checkbox"/> Indexed <input type="checkbox"/> Sequential <input type="checkbox"/> Direct
Record Size (Characters) :	200
Block Size :	4000
Number of Records : Maximum	45,000
Average :	42,000
Percent Growth per Year :	6 %
Data Structure	เรคคอร์ดลูกค้า
Primary Key	หมายเลขลูกค้า
Secondary Keys	ชื่อลูกค้า
Comments	แฟ้มข้อมูลหลักลูกค้านี้ จะถูกคัดลอกไปไว้ในแฟ้มประวัติลูกค้าและถูกปรับปรุง ถ้าหากลูกค้าไม่เคยซื้อสินค้าเลยภายในระยะเวลา 5 ปี

Components of a Traditional Analysis Model

