

## บทที่ 4 แผนภาพกระแสข้อมูล (Data Flow Diagram)

### วัตถุประสงค์

- ทราบความสำคัญและวัตถุประสงค์ของแผนภาพกระแสข้อมูล (Data Flow Diagram)
- เข้าใจสัญลักษณ์ต่างๆ และขั้นตอนการเขียนแผนภาพกระแสข้อมูล
- สามารถวิเคราะห์และเขียนแผนภาพกระแสข้อมูลได้

## บทที่ 4 แผนภาพกระแสข้อมูล (Data Flow Diagram)

จากการที่นักวิเคราะห์ระบบได้ทำการรวบรวมข้อมูลต่างๆ มาเรียบร้อยแล้วนั้น ข้อมูลที่ได้มาอาจอยู่ในรูปแบบเอกสาร แบบฟอร์ม รายงาน หรือบทสรุปการสัมภาษณ์ เป็นต้น ขั้นตอนต่อไปเป็นขั้นตอนของการวิเคราะห์ระบบงาน โดยจะเริ่มด้วยการวิเคราะห์ระบบงานเดิมที่ดำเนินการอยู่ในปัจจุบัน เพื่อจะได้ทราบถึงขั้นตอนการปฏิบัติงานปัจจุบัน และทำการสร้างแบบจำลองเชิงตรรกะ (Logical Model) ใหม่เรียกว่า แผนภาพกระแสข้อมูล (Data Flow Diagram) ซึ่งแบบจำลองนี้เปรียบเสมือนแบบบ้าน หรือแบบพิมพ์เขียว (Blueprint) ที่แสดงให้เห็นเจ้าของระบบงานเห็นแบบพิมพ์ของระบบงานก่อนว่า มีรายละเอียดของระบบงานของตนเองอย่างไร ตรงกับความต้องการของตนหรือไม่ รวมถึงผู้ที่นำแบบพิมพ์นี้ไปสร้างเป็นระบบงาน ก็จะได้ทำการสร้างได้ถูกต้อง เปรียบเสมือนการสร้างบ้านโดยมีแบบแปลนนั่นเอง

### 4.1 ความหมายของแผนภาพกระแสข้อมูล (Data Flow Diagram)

แผนภาพกระแสข้อมูล (Data Flow Diagram: DFD) หรือเรียกอีกอย่างหนึ่งว่า แผนภาพการไหลของข้อมูล เป็นเครื่องมือที่ใช้เพื่อแสดงการไหลของข้อมูลและการประมวลผลต่างๆ ในระบบ สัมพันธ์กับแหล่งเก็บข้อมูลที่ใช้ โดยแผนภาพนี้จะเป็นสื่อที่ช่วยให้การวิเคราะห์เป็นไปได้อย่างง่ายดาย และมีความเข้าใจตรงกันระหว่างผู้วิเคราะห์ระบบเอง หรือระหว่างผู้วิเคราะห์ระบบกับโปรแกรมเมอร์ หรือระหว่างผู้วิเคราะห์ระบบกับผู้ใช้งาน (รัชนี กัลยาวิทย์ และ อัจฉรา ธารอุไรกุลม 2542)

ประโยชน์ที่ได้จากการใช้แผนภาพกระแสข้อมูล มีดังนี้

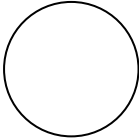
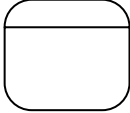
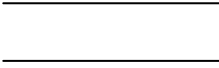
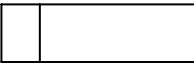




1. มีความอิสระในการใช้งาน โดยไม่ต้องมีเทคนิคอื่นมาช่วย เนื่องจากสามารถใช้สัญลักษณ์ต่างๆ แทนสิ่งที่จะวิเคราะห์มา
2. เป็นสื่อที่ง่ายต่อการแสดงความสัมพันธ์ระหว่างระบบใหญ่และระบบย่อย ซึ่งจะทำให้เข้าใจความสัมพันธ์ต่างๆ ได้ดี
3. เป็นสื่อที่ช่วยในการวิเคราะห์ระบบให้เป็นไปได้อย่างง่ายดาย และมีความเข้าใจตรงกันระหว่างผู้วิเคราะห์ระบบเอง หรือระหว่างผู้วิเคราะห์ระบบกับโปรแกรมเมอร์ หรือระหว่างผู้วิเคราะห์ระบบกับผู้ใช้งาน
4. ช่วยในการวิเคราะห์ระบบให้สะดวก โดยสามารถเห็นข้อมูลและขั้นตอนต่างๆ เป็นแผนภาพ

### 4.2 สัญลักษณ์ที่ใช้ในการเขียนแผนภาพกระแสข้อมูล

สัญลักษณ์ที่ใช้ในการเขียนแผนภาพกระแสข้อมูลนั้น ประกอบด้วย 4 สัญลักษณ์ ที่แสดงถึงการประมวลผลการไหลของข้อมูล ส่วนที่ใช้เก็บข้อมูล และสิ่งที่อยู่นอกระบบ โดยได้มีการศึกษาคิดค้นพัฒนาวิธีการอยู่หลายแบบ แต่ที่เป็นมาตรฐานมี 2 กลุ่ม คือกลุ่มที่คิดค้นโดย Gane and Sarson (1979) และ กลุ่มของ DeMarco and Yourdon

(SeMarco, 1979) ถึงแม้สัญลักษณ์บางอย่างของสององค์กรนี้จะต่างกัน แต่องค์ประกอบของแผนภาพและหลักการเขียนแผนภาพไม่ได้แตกต่างกัน ดังตารางที่ 4.1

**ตารางที่ 4.1** สัญลักษณ์ต่างๆ ที่ใช้ในการเขียนแผนภาพการไหลของข้อมูล

ชื่อสัญลักษณ์	DeMarco & Yourdon symbols	Gane & Sarson symbols
การประมวลผล (Process)		
แหล่งเก็บข้อมูล (Data Store)		
กระแสข้อมูล (Data Flow)		
สิ่งที่อยู่ภายนอก (External Entity)		

ในหนังสือเล่มนี้จะใช้สัญลักษณ์ในการเขียน DFD ตามแนวคิดของกลุ่ม Gane and Sarson

เนื่องจากสัญลักษณ์การเขียนแผนภาพกระแสข้อมูล ประกอบด้วยสัญลักษณ์ต่างๆ ซึ่งมีความหมายในตัวเอง ดังนั้นการเขียนแผนภาพกระแสข้อมูลจึงต้องมีกฎเกณฑ์ต่างๆ เพื่อแสดงถึงความถูกต้องในการเขียนแผนภาพดังนี้

#### 4.2.1. สัญลักษณ์การประมวลผล (Process Symbol)

การประมวลผล (Process) เป็นการเปลี่ยนแปลงข้อมูลจากรูปแบบหนึ่ง (Input) ไปเป็นอีกรูปแบบหนึ่ง (Output) เช่น การคำนวณรายได้สุทธิของลูกค้าจากรายวัน จะต้องประกอบด้วยข้อมูลนำเข้าที่เป็น "อัตราค่าจ้างต่อชั่วโมง" และ "จำนวนชั่วโมงการทำงาน" เมื่อผ่านการประมวลผลแล้วจะได้ "รายได้สุทธิ"

ตัวอย่างการประมวลผล ได้แก่

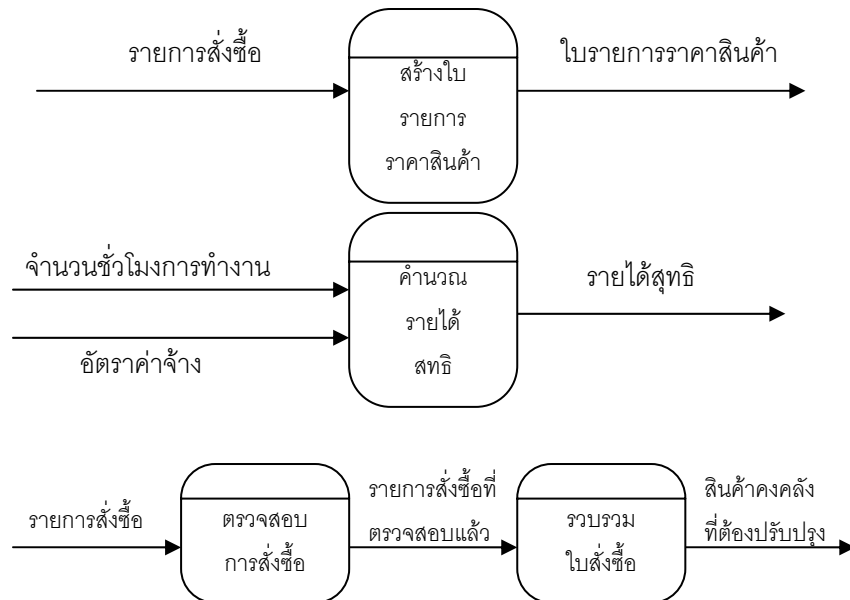
- คำนวณค่าคอมมิชชั่น
- ตรวจสอบใบสั่งซื้อ
- ลงทะเบียน เป็นต้น

การใช้สัญลักษณ์การประมวลผล

1. ต้องใช้สัญลักษณ์การประมวลผล (Process) คู่กับสัญลักษณ์ กระแสข้อมูล (Data Flow) เสมอ โดยที่ถ้าลูกศรชี้เข้าหมายถึงเป็นข้อมูลนำเข้า ถ้าลูกศรชี้ออกหมายถึงเป็นข้อมูลออกจากการประมวลผล ซึ่ง 1 Process สามารถมีข้อมูลนำเข้ามากกว่า 1 เส้น หรือข้อมูลออกมากกว่า 1 เส้นได้

2. การตั้งชื่อของ Process ควรเป็นวลีเดียวที่อธิบายการทำงานทั้งหมดได้ และควรอธิบายการทำงานอย่างใดอย่างหนึ่งโดยเฉพาะมากกว่าที่จะอธิบายการทำงานอย่างกว้างๆ เช่น หากแสดงถึงการประมวลผล “ตรวจสอบรายการ” ควรจะระบุว่าเป็น “การตรวจสอบรายการถอนเงิน” หรือ “ตรวจสอบรายการค่าใช้จ่ายรายสัปดาห์” เป็นต้น

3. แต่ละ Process จะมีแต่ข้อมูลเข้าอย่างเดียว หรือออกอย่างเดียวนิไม่ได้



รูปที่ 4.1 ตัวอย่างการใช้สัญลักษณ์การประมวลผล (Process)

#### 4.2.2. สัญลักษณ์กระแสข้อมูล (Data Flow Symbol)

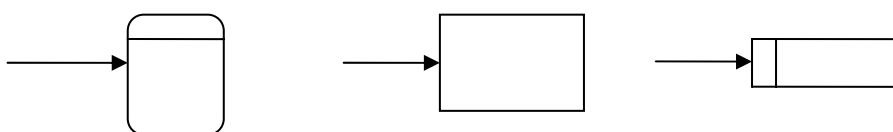
กระแสข้อมูล (Data Flow) เป็นเส้นทางในการไหลของข้อมูลจากส่วนหนึ่ง ไปยังอีกส่วนหนึ่งของระบบสารสนเทศ โดยจะมีลูกศรแสดงถึงการไหลจากปลายลูกศร ไปยังหัวลูกศร ซึ่งข้อมูลที่ปรากฏบนเส้นนี้จะเป็นได้ทั้งข้อความ ตัวเลข รายการเรคคอร์ดที่ระบบคอมพิวเตอร์สามารถนำไปประมวลผลได้

ตัวอย่างกระแสข้อมูล ได้แก่

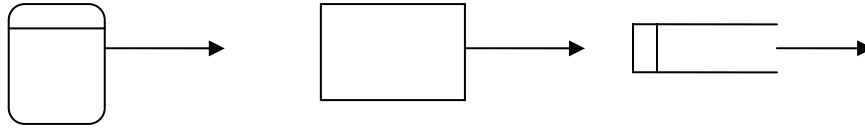
- ใบสั่งซื้อสินค้า
- ใบเสร็จรับเงิน
- เกรดของนักศึกษา
- ใบส่งของที่ผ่านการตรวจสอบแล้ว เป็นต้น

การใช้สัญลักษณ์กระแสข้อมูล

1. กระแสข้อมูลสามารถใช้คู่กับการประมวลผล (Process), สิ่งที่อยู่นอกระบบ (External Entities) หรือแหล่งเก็บข้อมูล (Data Store) ก็ได้ ขึ้นอยู่กับระบบงานว่า ข้อมูลนั้นจะนำไปไว้ที่ไหน หรือข้อมูลนั้นจะนำออกจากส่วนใด



รูปที่ 4.2 ก. ตัวอย่างการใช้สัญลักษณ์กระแสข้อมูล (Data Flow)



รูปที่ 4.2 ข. ตัวอย่างการใช้สัญลักษณ์กระแสข้อมูล (Data Flow)

2. การตั้งชื่อกระแสข้อมูล โดยทั่วไปจะตั้งชื่อด้วยคำเพียงคำเดียว ที่มีความหมายชัดเจนและเข้าใจง่าย ควรกำกับชื่อบนเส้นด้วย คำนาม เช่น “เวลาทำงาน”, “ใบสั่งซื้อสินค้า” เป็นต้น

3. ควรตั้งชื่อกระแสข้อมูล ตามข้อมูลที่ได้เปลี่ยนแปลงไปแล้วหลังจากออกจากการประมวลผล เนื่องจากการประมวลผลหรือ Process ใช้แสดงถึงการเปลี่ยนข้อมูล หรือการส่งผ่านข้อมูล ดังนั้น Data Flow ที่ออกจาก Process มักจะมีการเขียนชื่อกำกับให้แตกต่างออกไปจาก Data Flow ที่เข้ามาใน Process เสมอ

#### 4.2.3. สัญลักษณ์แหล่งเก็บข้อมูล (Data Store Symbol)

แหล่งที่เก็บข้อมูล (Data Store) เป็นส่วนที่ใช้แทนชื่อแฟ้มข้อมูลที่เก็บข้อมูล เพราะมีการประมวลผลหลายแบบที่จะต้องมีการเก็บข้อมูลไว้เพื่อที่จะได้นำไปใช้ภายหลัง ซึ่งแหล่งเก็บข้อมูลจะต้องมีทั้งข้อมูลเข้าและข้อมูลออก โดยข้อมูลที่ออกจากแหล่งเก็บข้อมูลจะอยู่ในลักษณะที่ถูกอ่านขึ้นมา ส่วนข้อมูลที่ไหลเข้าสู่แหล่งเก็บข้อมูลจะอยู่ในรูปของการบันทึก การเพิ่ม-ลบ แก้ไข

ตัวอย่างแหล่งเก็บข้อมูล ได้แก่

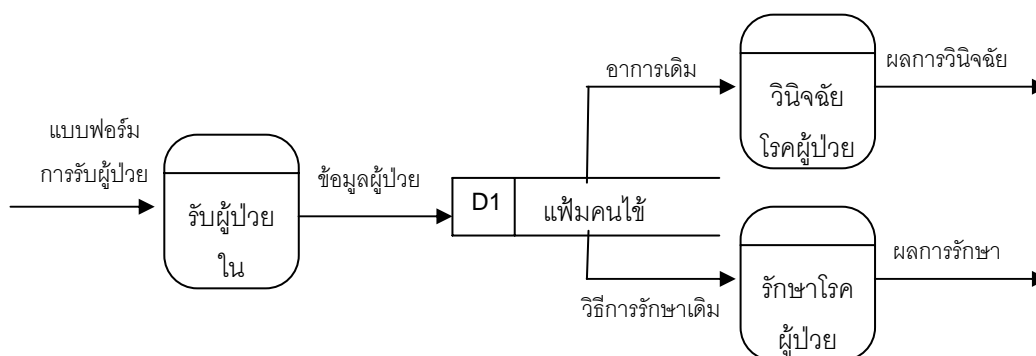
- แฟ้มคนไข้
- แฟ้มพนักงาน เป็นต้น

การใช้สัญลักษณ์แหล่งเก็บข้อมูล

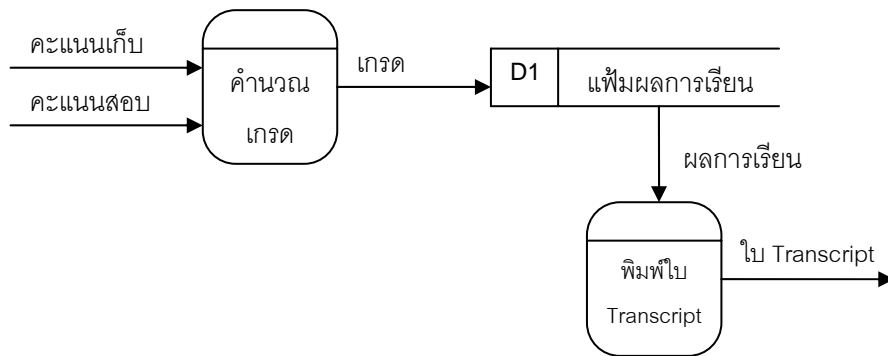
1. ต้องใช้สัญลักษณ์แหล่งเก็บข้อมูล (Data Store) คู่กับสัญลักษณ์ กระแสข้อมูล (Data Flow) เสมอ โดยที่ถ้าลูกศรชี้เข้าหมายถึง เป็นข้อมูลนำเข้าเก็บยังแหล่งเก็บ ถ้าลูกศรชี้ออกหมายถึง อ่านข้อมูลจากแหล่งเก็บข้อมูลไปใช้ในการประมวลผล

2. Data Store ต้องเชื่อมต่อการประมวลผล (Process) เสมอโดยเชื่อมผ่านกระแสข้อมูล (Data Flow)

3. เนื่องจาก Data Store ใช้แทนสิ่งที่เก็บข้อมูลเกี่ยวกับคน, สถานที่, หรือสิ่งของ ดังนั้นควรเขียนชื่อกำกับด้วยคำนาม เช่น “แฟ้มข้อมูลสินค้า”, “แฟ้มเวลาทำงานของพนักงาน” เป็นต้น



รูปที่ 4.3 ก. ตัวอย่างการใช้สัญลักษณ์แหล่งที่เก็บข้อมูล (Data Store)



รูปที่ 4.3 ข. ตัวอย่างการใช้สัญลักษณ์แหล่งที่เก็บข้อมูล (Data Store)

4. ใช้ตัวย่อ D1, D2 เป็นต้น เขียนด้านซ้ายมือของสัญลักษณ์ เพื่อแสดงว่าเป็นแหล่งเก็บข้อมูลอื่นที่เท่าใด สามารถเขียนซ้ำในระดับต่างๆ ของแผนภาพกระแสข้อมูลได้

5. Data Store ใช้แทนสิ่งที่เป็นที่เก็บข้อมูล ซึ่งอาจเป็นการทำด้วยมือ หรือเก็บในรูปแบบคอมพิวเตอร์คือ แฟ้มข้อมูลหรือฐานข้อมูล ก็ได้

#### 4.2.4. สัญลักษณ์สิ่งที่อยู่ภายนอก (External Entity Symbol)

สิ่งที่อยู่ภายนอก (External Entity) เป็นส่วนที่ใช้แทนคน แผนกภายในองค์กร และแผนกภายนอกองค์กร หรือระบบสารสนเทศอื่นที่เป็นส่วนที่จะให้ข้อมูลหรือรับข้อมูล

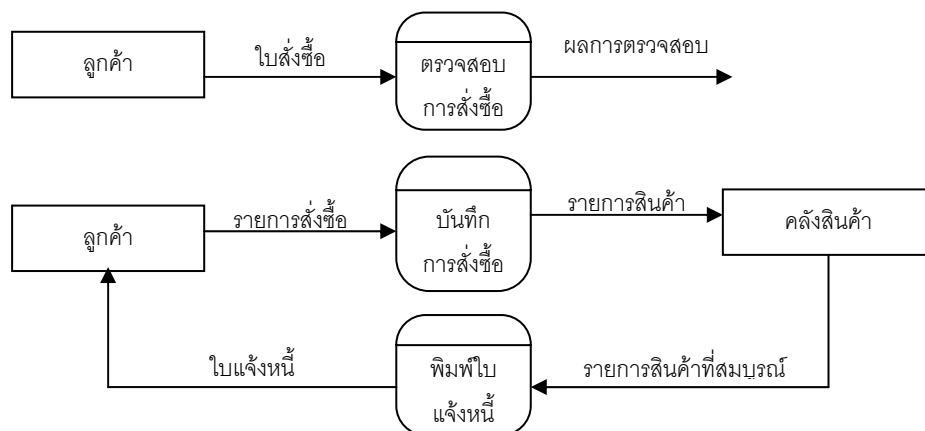
สิ่งที่อยู่ภายนอกแบบนี้ใช้แสดงถึงขอบเขตของระบบสารสนเทศ และแสดงถึงว่าระบบที่ศึกษาอยู่นี้จะติดต่อกับสิ่งที่อยู่ภายนอกด้วยวิธีใด (นำข้อมูลเข้ามา หรือได้ข้อมูลออกไป)

ตัวอย่างสิ่งที่อยู่ภายนอก ได้แก่

- นักศึกษา
- สินค้าคงคลัง เป็นต้น
- สมาชิก เป็นต้น

การใช้สัญลักษณ์สิ่งที่อยู่ภายนอก

1. ใช้สัญลักษณ์สิ่งที่อยู่ภายนอกคู่กับสัญลักษณ์กระแสข้อมูลเสมอ โดยที่ถ้าลูกศรชี้เข้า หมายถึง เป็นการนำข้อมูลจากหน่วยงานภายนอกเข้าสู่ระบบ ถ้าลูกศรชี้ออก หมายถึง ส่งข้อมูลจากระบบไปให้หน่วยงานภายนอก



รูปที่ 4.4 ตัวอย่างการใช้สัญลักษณ์สิ่งที่อยู่ภายนอก (External Entities)

### 4.3 กฎที่ใช้ในการเขียนแผนภาพกระแสข้อมูล

ในการเขียน Data Flow Diagram มีข้อแนะนำในการเขียนแผนภาพให้ถูกต้องดังนี้

ตารางที่ 4.2 กฎที่ใช้สัญลักษณ์ในการเขียนแผนภาพกระแสข้อมูล

ผิด	ถูก	คำอธิบาย
<b>การประมวลผล(Process)</b>		
		ในการประมวลผลจะมีข้อมูลเข้าอย่างเดียวนำเข้าไม่ได้
		ในการประมวลผลจะมีข้อมูลออกอย่างเดียวนำออกไม่ได้
		ข้อความที่บรรจุอยู่ในการประมวลผลจะต้องเป็นคำกริยา หรือ คำนามที่แสดงถึงกริยาเท่านั้น
<b>แหล่งเก็บข้อมูล(Data Store)</b>		
		ไม่สามารถส่งข้อมูลโดยตรงจาก แหล่งเก็บข้อมูลอันหนึ่งไปยัง แหล่งเก็บข้อมูลอีกอันได้โดยตรง ข้อมูลต้องผ่านการจาก ประมวลผลจึงจะไป ที่ แหล่งเก็บข้อมูลได้
		ไม่สามารถส่งข้อมูลจากสิ่งที่อยู่ภายนอก(External Entity)ไปยัง แหล่งเก็บข้อมูลได้โดยตรง ข้อมูลต้องผ่านการจากการประมวลผล ที่ ได้รับข้อมูลจากแหล่งข้อมูลภายนอกแล้วจึงส่งต่อให้ แหล่งเก็บ ข้อมูล
		ไม่สามารถส่งข้อมูลจากแหล่งเก็บข้อมูลไปยังสิ่งที่อยู่ภายนอก ระบบปลายทางได้โดยตรง ต้องส่งผ่าน การประมวลผลไปยังสิ่งที่ อยู่ภายนอกระบบ
		ข้อความที่บรรจุอยู่ในแหล่งเก็บข้อมูลต้องเป็นคำนาม
<b>กระแสข้อมูล(Data Flow)</b>		
		กระแสข้อมูล จะมีทิศทางไหลของข้อมูลเพียงทิศเดียวใน 1 กระแสข้อมูล เนื่องจากไม่มีการทำงานใดที่เกิดขึ้นพร้อมกัน

ผิด	ถูก	คำอธิบาย
<b>กระแสข้อมูล(Data Flow) ต่อก</b>		
		การแยกของข้อมูลออกจาก เส้นกระแสข้อมูล ไปสู่ การประมวลผล อื่น ๆ พร้อมกัน ต้องเป็นข้อมูลเดียวกัน
		การเชื่อมข้อมูลจากข้อมูลมากกว่า 1 แหล่งเข้าสู่ เส้นกระแสข้อมูล เดียวกันต้องเป็นข้อมูลเดียวกัน
		กระแสข้อมูลไม่สามารถไหลกลับเข้าสู่การประมวลผลเดิมได้โดยตรง จะต้องมีการประมวลอย่างน้อยหนึ่งอย่างมาจัดการกับข้อมูลดังกล่าวก่อน สร้างกระแสข้อมูลอื่น แล้วจึงคืนค่ากระแสข้อมูลเดิมกลับมาเพื่อการประมวลผลเริ่มต้น
	รหัสนักศึกษา 	ข้อความที่อยู่บน Data Flow จะเป็นค่านาม
<b>สิ่งที่อยู่ภายนอก(External Entity)</b>		
		ไม่สามารถส่งข้อมูลโดยตรงจาก สิ่งที่อยู่ภายนอกระบบอันหนึ่งไปยังสิ่งที่อยู่ภายนอกระบบอีกอันได้โดยตรง ข้อมูลต้องผ่านการจากประมวลผลภายในระบบก่อนจึงส่งไปยังสิ่งที่อยู่ภายนอก ระบบภายหลังได้
		ข้อความที่บรรจุอยู่ในสิ่งที่อยู่ภายนอกระบบต้องเป็นค่านาม

#### 4.4 ประเภทของแผนภาพกระแสข้อมูล

การเขียน DFD นั้นสามารถเขียนได้ 2 แบบ คือแบบตรรกะ (Logical Data Flow Diagram) และแบบการภาพ (Physical Data Flow Diagram)

##### 4.4.1 แผนภาพกระแสข้อมูลแบบตรรกะ (Logical DFD)

แผนภาพนี้จะเป็นการเน้นในส่วนของผู้กิจ ว่าผู้กิจมีการทำงานอย่างไร มีเหตุการณ์อะไรบ้างที่เกิดขึ้น, ข้อมูลที่ต้องการมีอะไรบ้าง และได้ข้อมูลอะไรจากเหตุการณ์นั้นๆ แต่ไม่ได้บอกว่าระบบจะถูกสร้างอย่างไร



#### 4.4.2 แผนภาพกระแสข้อมูลแบบกายภาพ (Physical DFD)

แผนภาพนี้จะบอกได้ว่าจะสร้างระบบอย่างไร ประกอบด้วยฮาร์ดแวร์, ซอฟต์แวร์, แฟ้มข้อมูล และบุคคลใดที่เกี่ยวข้องบ้างในระบบ

**ตารางที่ 4.3** ข้อแตกต่างระหว่างแผนภาพกระแสข้อมูลแบบตรรกะ กับแบบกายภาพ

รูปแบบการออกแบบ	แบบตรรกะ (Logical DFD)	แบบกายภาพ (Physical DFD)
ขั้นตอนการประมวลผล	<ul style="list-style-type: none"> <li>- แสดงถึง ธุรกิจสามารถที่จะดำเนินการได้อย่างไร</li> <li>- เป็นกิจกรรมทางธุรกิจ</li> </ul>	<ul style="list-style-type: none"> <li>- แสดงถึง ระบบสามารถที่จะทำงานได้อย่างไร</li> <li>- เป็นโปรแกรม หรือโมดูล หรือขั้นตอนการทำงานที่ทำด้วยมือ</li> </ul>
แหล่งเก็บข้อมูล	- เป็นแหล่งที่เก็บรวบรวมข้อมูลลักษณะใดก็ได้	- เป็น Physical File, Database และ Manual File
ชนิดของแหล่งเก็บข้อมูล	- เป็นการเก็บข้อมูลแบบถาวร	- เป็นการเก็บข้อมูลที่แยกเป็นแฟ้มหลักและแฟ้มรายการ ต้องการประมวลผลในเวลาที่แตกต่างกัน สามารถเชื่อมต่อกันได้โดยใช้แหล่งเก็บข้อมูล
การควบคุมระบบ	- เป็นการควบคุมที่เกี่ยวข้องกับธุรกิจ	- แสดงถึง การควบคุมสำหรับการป้อนข้อมูลที่ถูกต้อง

#### ประโยชน์ของการสร้างแผนภาพกระแสข้อมูลแบบตรรกะ (Logical DFD)

1. ช่วยให้การสื่อสารระหว่างผู้ใช้งานให้ดีขึ้น
2. ช่วยให้ระบบมีความมั่นคงมากขึ้น
3. ช่วยให้วิศวกรระบบมีความเข้าใจกับการดำเนินงานของระบบได้ชัดเจน
4. ช่วยในการบำรุงรักษาและมีความยืดหยุ่นมากขึ้น
5. ช่วยลดความซับซ้อน และง่ายต่อการสร้างแผนภาพกระแสข้อมูลแบบกายภาพต่อไป

#### ประโยชน์ของการสร้างแผนภาพกระแสข้อมูลแบบกายภาพ (Physical DFD)

1. ขั้นตอนการประมวลผลนั้นเป็นการทำงานด้วยระบบมือ หรือระบบอัตโนมัติ
2. แผนภาพแบบกายภาพจะอธิบายรายละเอียดขั้นตอนการประมวลผลได้ละเอียด
3. มีการแสดงถึงลำดับขั้นตอนการทำงาน

## 4.5 การพัฒนาแผนภาพกระแสข้อมูล

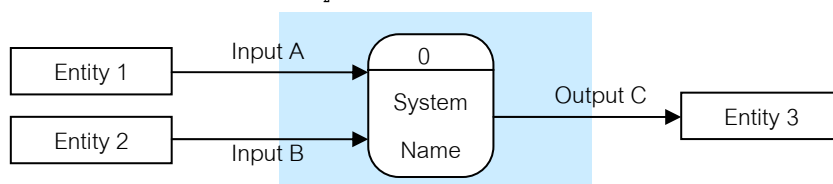
ในการเขียนแผนภาพกระแสข้อมูลนั้น ควรเขียนให้เป็นระบบงาน ซึ่งสามารถสรุปขั้นตอนของการเขียนให้สมบูรณ์ได้ดังนี้

1. กำหนดรายการกิจกรรมต่างๆ ของธุรกิจ และแยกออกมาว่าอยู่ในรูปแบบใด (External Entities หรือ Data Flows หรือ Processes หรือ Data Stores)
2. สร้างแผนภาพระดับสูงสุด (Context Diagram) ซึ่งแสดงถึง External Entities และข้อมูลที่ไหลเข้าและออกจากระบบหลัก โดยไม่สนใจแหล่งเก็บข้อมูล
3. เขียน Diagram ระดับถัดไป คือแผนภาพระดับล่าง (เรียกว่า Diagram 0 หรือ Parent Diagram) ซึ่งแสดงถึง Process ต่างๆ ที่มี แต่เขียนให้อยู่ในรูปแบบทั่วไป พร้อมกันนั้นให้แสดง Data Store ในระดับนี้ด้วย
4. สร้างแผนภาพระดับลูกของแต่ละ Process ใน Diagram 0 เรียกแผนภาพในระดับนี้ว่า แผนภาพกระแสข้อมูลระดับ 1 (Level-1 diagram) ถ้าหากมีรายละเอียดของการทำงานย่อยจากระดับนี้อีก ก็ให้แตกรายละเอียดลงไปจนกระทั่งสิ้นสุด ส่วนชื่อของระดับก็จะเป็น Level-2 diagram , Level-3 diagram ไปเรื่อยๆ จนกระทั่งหมด
5. ตรวจสอบหาข้อผิดพลาด และดูว่าคำกำกับบนเส้น Data Flow แต่ละเส้น รวมถึง Process แต่ละอันนั้นสื่อความหมายหรือไม่
6. หลังจากเขียนแผนภาพจนครบทุกการทำงานแล้ว ให้ทำการตรวจสอบสมดุลระหว่างข้อมูลเข้าและข้อมูลออกของแผนภาพ DFD กับ Context diagram
7. พัฒนารูปแบบใหม่ จาก Logical Data Flow Diagram ให้ไปอยู่ในรูป Physical Data Flow Diagram เพื่อแยกระหว่างระบบที่ทำด้วยมือ กับระบบที่ทำงานอัตโนมัติ
8. แบ่งส่วนของ Physical Data Flow Diagram โดยการแยกหรือแบ่งกลุ่มของ Diagram ออก เพื่อให้สามารถนำไปเขียนโปรแกรม หรือเพื่อการดำเนินการระบบได้

เมื่อนักวิเคราะห์ระบบรวบรวมข้อมูลที่ได้มาจากการสัมภาษณ์ การออกแบบสอบถาม และเทคนิคต่างๆ ในการรวบรวมข้อมูลจริงเกี่ยวกับระบบ ซึ่งเมื่อผ่านขั้นตอนเหล่านี้ นักวิเคราะห์ระบบจะต้องทราบว่า มีบุคคลใด หน่วยงานใด ข้อมูลใด การประมวลผลอะไรเข้ามาเกี่ยวข้องกันบ้าง ซึ่งเมื่อถึงจุดนี้นักวิเคราะห์ระบบก็ควรพร้อมที่จะสร้างโมเดลของระบบงานในรูปแบบกราฟิก

### 4.5.1 แผนภาพกระแสข้อมูลระดับสูงสุด (Context Diagram)

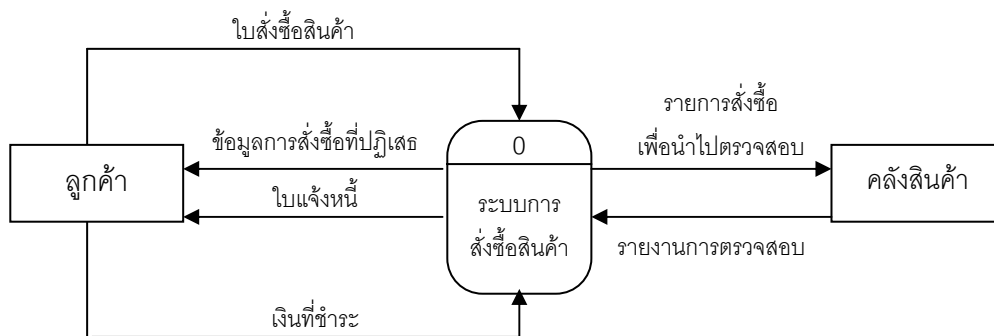
แผนภาพกระแสข้อมูลระดับสูงสุด เป็นแผนภาพที่แสดงถึงขอบเขตของสารสนเทศนั้น โดยจะเป็นมุมมองระดับสูงว่ามีหน่วยงานใดเกี่ยวข้องบ้าง ติดต่อกับระบบโดยมีการรับและส่งข้อมูลใดกับระบบ ซึ่งแผนภาพระดับนี้จะยังไม่กล่าวถึง สัญลักษณ์การเก็บข้อมูล (Data Store Symbol)



รูปที่ 4.5 รูปแบบการเขียน Context Diagram (ที่มา: Kenneth E.Kendall and Julie E.Kendall, 2002: 246)

### หลักการสร้าง Context Diagram ที่ดี

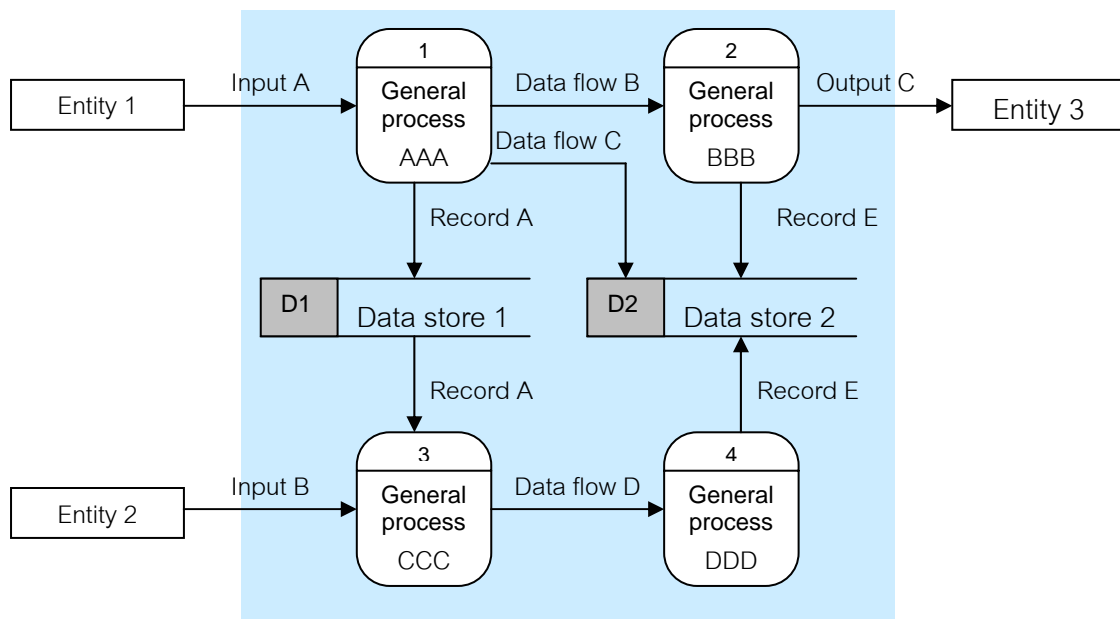
1. ควรจะเขียนให้ครอบคลุมระบบ ให้อยู่ใน 1 หน้ากระดาษ
2. ชื่อของ Process ควรเป็นชื่อของระบบสารสนเทศ นั้น
3. ชื่อที่เขียนกำกับสัญลักษณ์ต่างๆ จะต้องเขียนโดยไม่ให้ชื่อซ้ำกัน ถ้าสัญลักษณ์นั้นแทนสิ่งที่แตกต่างกัน
4. ควรหลีกเลี่ยงเส้นที่จะต้องเขียนคร่อมเส้นกัน
5. ในการเขียนชื่อย่อ จะต้องเขียนโดยบันทึกความหมายของตัวย่อนั้น ใน พจนานุกรมข้อมูล (Data Dictionary)



รูปที่ 4.6 ตัวอย่างการเขียน Context Diagram ของระบบการสั่งซื้อสินค้า

#### 4.5.2 แผนภาพกระแสข้อมูลระดับล่าง (Diagram 0 หรือ Parent Diagram)

แผนภาพกระแสข้อมูลระดับล่าง เป็นแผนภาพที่ให้รายละเอียดเพิ่มเติมจากแผนภาพระดับสูงสุด ทำให้เห็นภาพรวมของแผนภาพกระแสข้อมูล ซึ่งจะมีรายละเอียดมากกว่าแผนภาพกระแสข้อมูลระดับสูงสุด

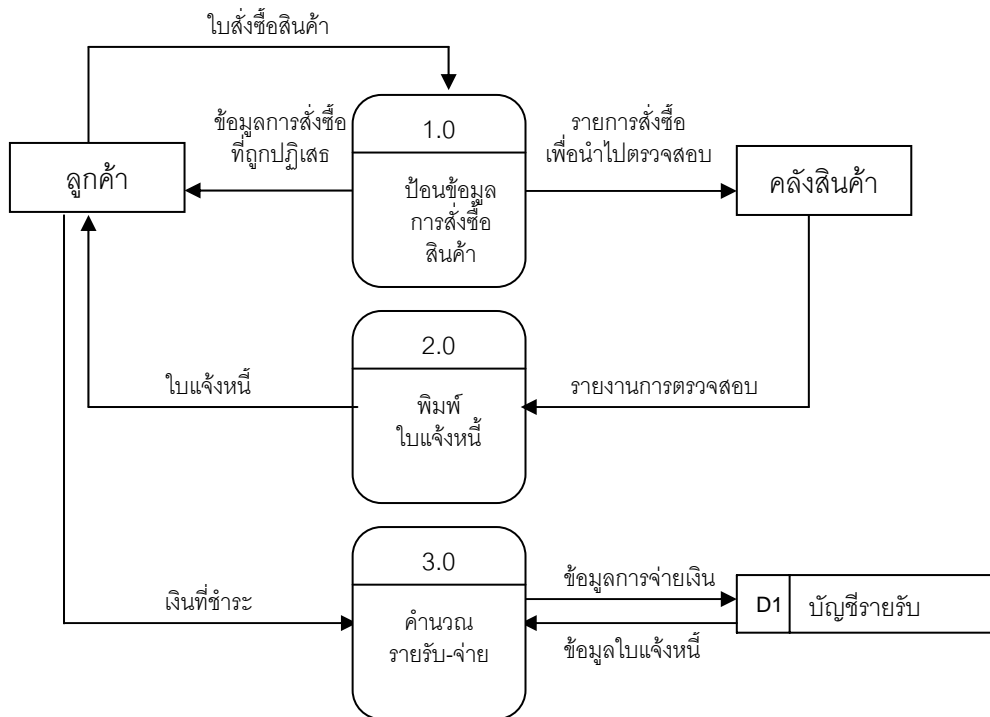


รูปที่ 4.7 รูปแบบการเขียน Diagram 0 (พิจารณาเปรียบเทียบกับรูปที่ 4.5)

(ที่มา: Kenneth E.Kendall and Julie E.Kendall, 2002: 246)

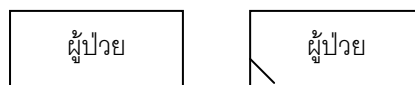
### หลักการสร้างแผนภาพกระแสข้อมูลระดับล่าง

1. การเขียนแผนภาพกระแสข้อมูลควรเขียนให้ดูง่ายและให้แสดงอยู่ในกระดาษแผ่นเดียว
2. ลูกศรไม่ควรทับหรือข้ามกัน
3. แต่ละ Process จะต้องมีความหมายเลขกำกับเป็นเลขจำนวนเต็ม โดยการลำดับหมายเลข Process ไม่ได้หมายถึงการทำงานต้องทำงานตามลำดับของ Process และ Process ไม่สามารถทำการซ้ำได้



รูปที่ 4.8 ตัวอย่างการเขียน Diagram 0 ของระบบการสั่งซื้อสินค้า (พิจารณาเปรียบเทียบกับรูปที่ 4.6)

4. External Entity ทั้งหมดของระบบ ที่เขียนใน Context diagram จะต้องแสดงอยู่ใน Diagram 0 ด้วย โดยที่สามารถทำการซ้ำได้ ด้วยเครื่องหมาย \ ตรงมุมล่างซ้ายของสัญลักษณ์ External Entity

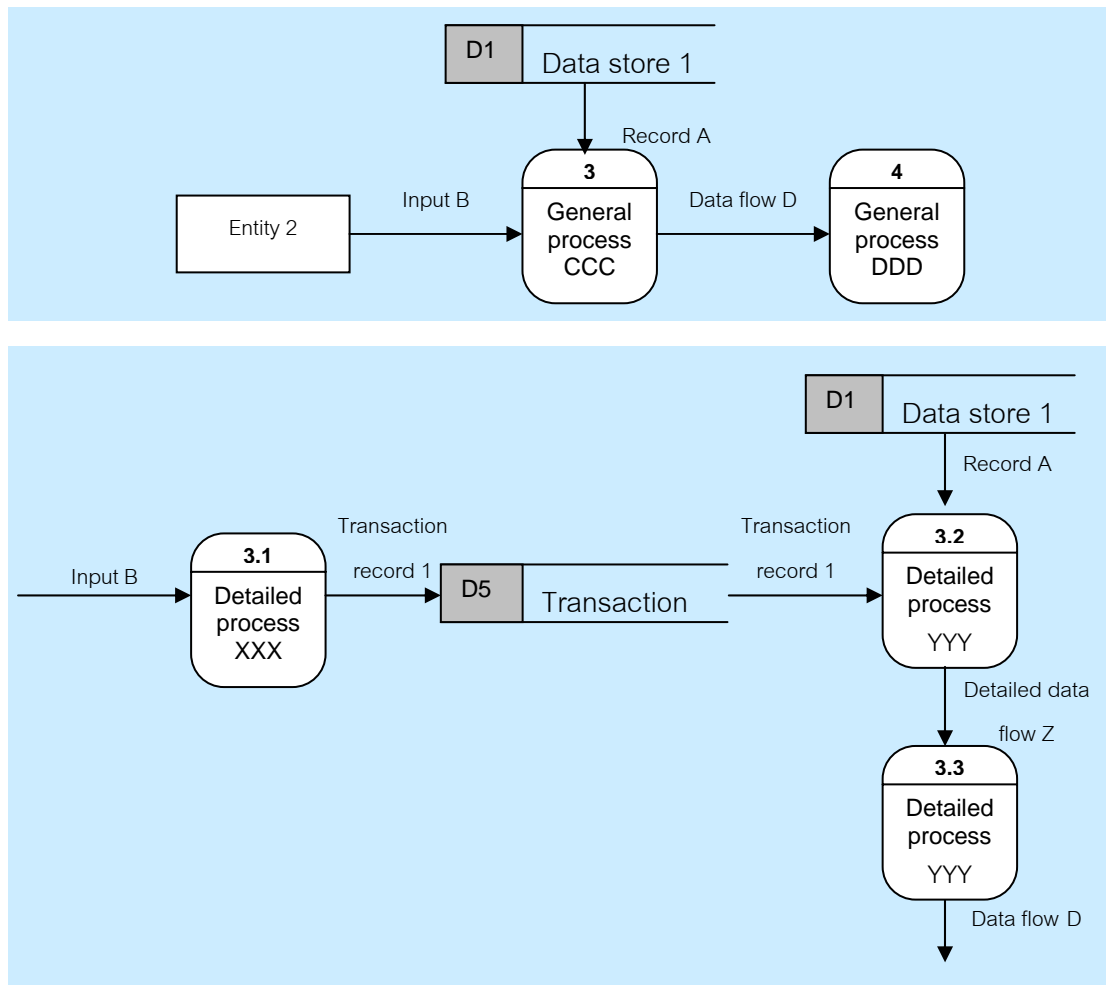


รูปที่ 4.9 สิ่งที่อยู่นอกกระบวนการและการทำซ้ำสิ่งที่อยู่นอกกระบวนการ

5. จำนวน Process ไม่ควรมีมากเกินไป หรือน้อยเกินไป จำนวน Process ที่มากเกินไปจะทำให้แผนภาพกระแสข้อมูลอ่านยาก และมีความซับซ้อนยิ่งขึ้น

#### 4.5.3 แผนภาพกระแสข้อมูลระดับต่ำ (Lower Level Data Flow Diagram)

แผนภาพกระแสข้อมูลระดับต่ำ เป็นแผนภาพที่มีรายละเอียดเพิ่มเติมจากแผนภาพระดับล่าง โดยในแผนภาพระดับนี้ จะมีการแตกการประมวลผลออกเป็นการประมวลผลย่อยๆ อีกได้

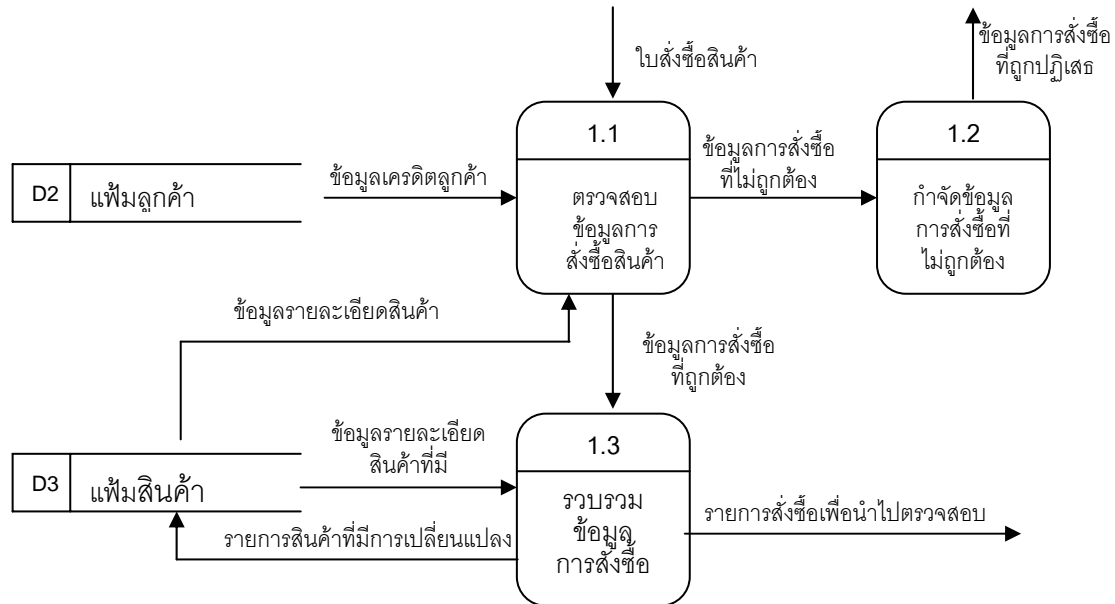


รูปที่ 4.10 รูปแบบการเขียน Child Diagram (Level-1) ของ Process 3.0 (พิจารณาเปรียบเทียบกับรูปที่ 4.7)  
(ที่มา: Kenneth E.Kendall and Julie E.Kendall, 2002: 248)

#### หลักการสร้างแผนภาพกระแสข้อมูลระดับต่ำ

1. แต่ละ Process บน Diagram 0 (หรือ Parent Process) สามารถนำมาสร้าง Diagram ย่อย (เรียกว่า Child Diagram) ได้
2. ในการสร้าง Child Diagram นั้น ข้อมูลที่เข้ามาและออกจาก Child Diagram จะต้องเป็นข้อมูลที่เข้ามาและออกจาก Parent Process
3. เมื่อมีการแยก Process ออกเป็น Child Diagram จะต้องมีการกำหนดหมายเลขลำดับให้กับ Child Diagram นั้น โดยจะตั้งตาม Parent Process (เช่น Process 3 แยกเป็น Diagram Level-1 ของ Process ที่ 3)
4. หมายเลขกำกับแต่ละ Process ใน Child Diagram ก็จะใช้หมายเลขขึ้นต้นเหมือน Parent Process แล้วตามด้วยจุดทศนิยม และหมายเลขย่อยลงไป (เช่น Parent Process เป็น Process 3 ดังนั้นใน Diagram 3 ก็จะมี Process 3.1, 3.2, 3.3 ไปเรื่อยๆ จนกว่าจะหมด)

5. โดยปกติแล้ว External Entity มักจะไม่แสดงบน Child Diagram ที่ต่ำกว่า Diagram 0
6. บน Child Diagram อาจมี Data Store ปรากฏขึ้นมา โดยที่ไม่มีใน Parent Process ได้
7. Process ต่างๆ อาจมีการแตกหรือไม่แตกเป็น Child Diagram ก็ได้ ขึ้นอยู่กับระดับความซับซ้อนของ Process นั้นๆ



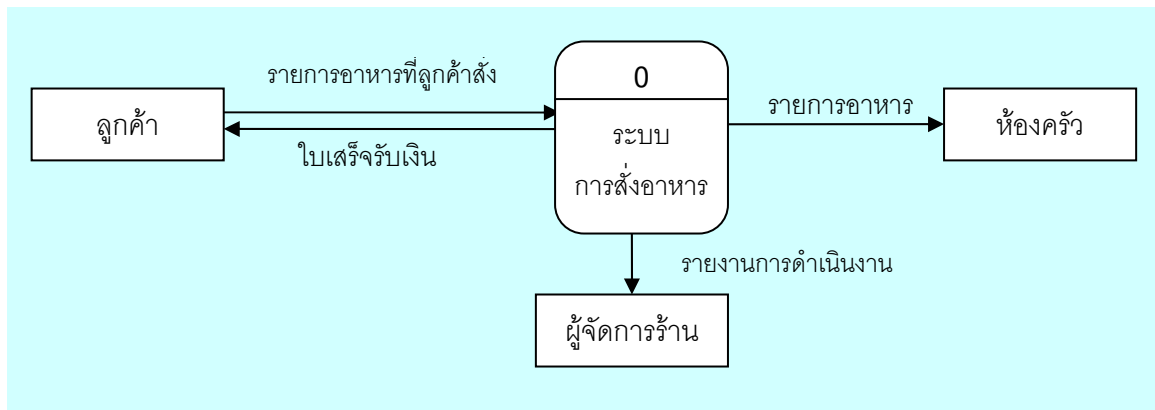
รูปที่ 4.11 ตัวอย่างการเขียน Child Diagram (Level-1) ของ Process 1.0-การป้อนข้อมูลการสั่งซื้อ  
(พิจารณาเปรียบเทียบกับรูปที่ 4.8)

#### 4.5.4 การตรวจสอบแผนภาพกระแสข้อมูลเพื่อหาข้อผิดพลาด

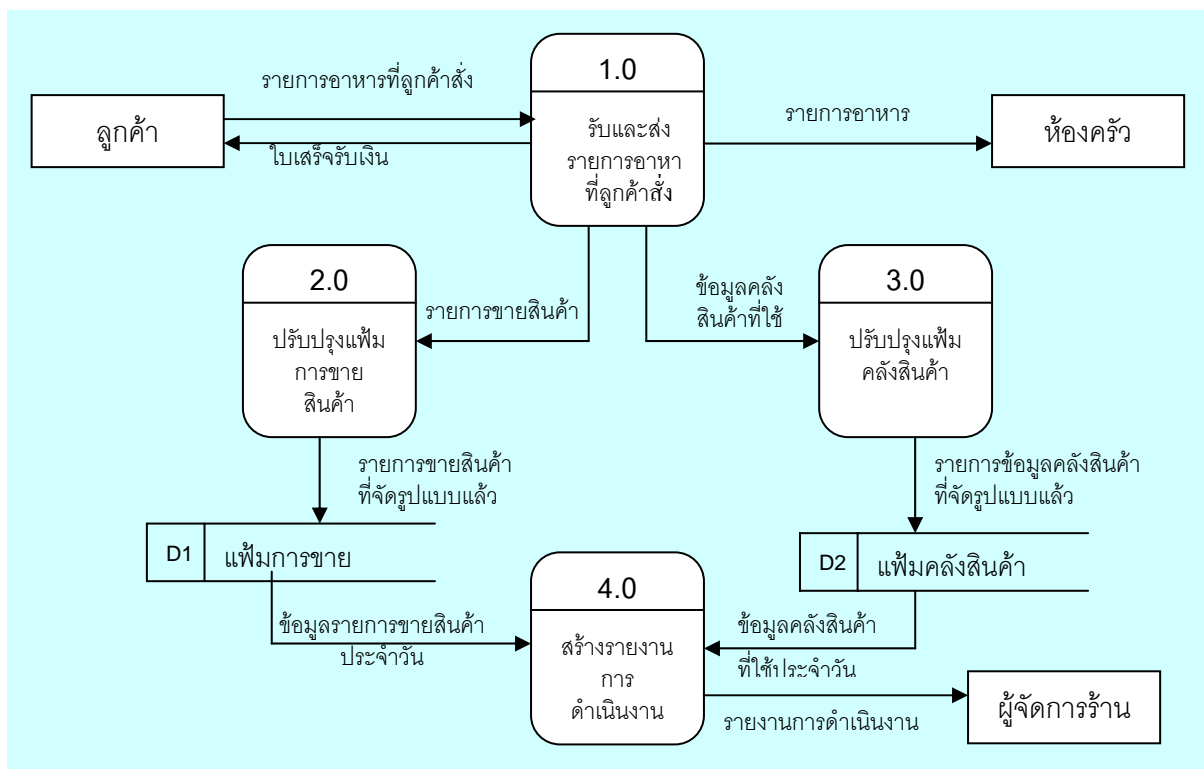
ในระหว่างการเขียนแผนภาพกระแสข้อมูลนั้น นักวิเคราะห์จะต้องเขียนแผนภาพกระแสข้อมูลโดยการกำหนดเป็นระดับต่างๆ ซึ่งสิ่งที่ต้องตรวจสอบในการเขียนแผนภาพกระแสข้อมูล คือ

1. ตรวจสอบว่า มีการใช้สัญลักษณ์ต่างๆ ถูกต้องตามกฎการเขียนแผนภาพกระแสข้อมูล ดังตารางที่ 4.3 หรือไม่
2. ตรวจสอบหาข้อผิดพลาด และดูว่าคำกำกับบนเส้น Data Flow แต่ละเส้น รวมถึง Process แต่ละอันนั้น สื่อความหมายหรือไม่
3. หลังจากเขียนแผนภาพจนครบทุกการทำงานแล้ว ให้ทำการตรวจสอบสมดุลระหว่างข้อมูลเข้าและข้อมูลออกของแผนภาพ นั่นคือ ตรวจสอบว่าแผนภาพระดับสูงสุด (Context Diagram) มีความสมดุลระหว่างข้อมูลเข้าและข้อมูลออก กับแผนภาพระดับล่าง (Diagram 0) หรือไม่ และตรวจสอบว่าแผนภาพระดับล่าง (Diagram 0) มีความสมดุลระหว่างข้อมูลเข้าและข้อมูลออก กับแผนภาพระดับต่ำ (Child Diagram) หรือไม่

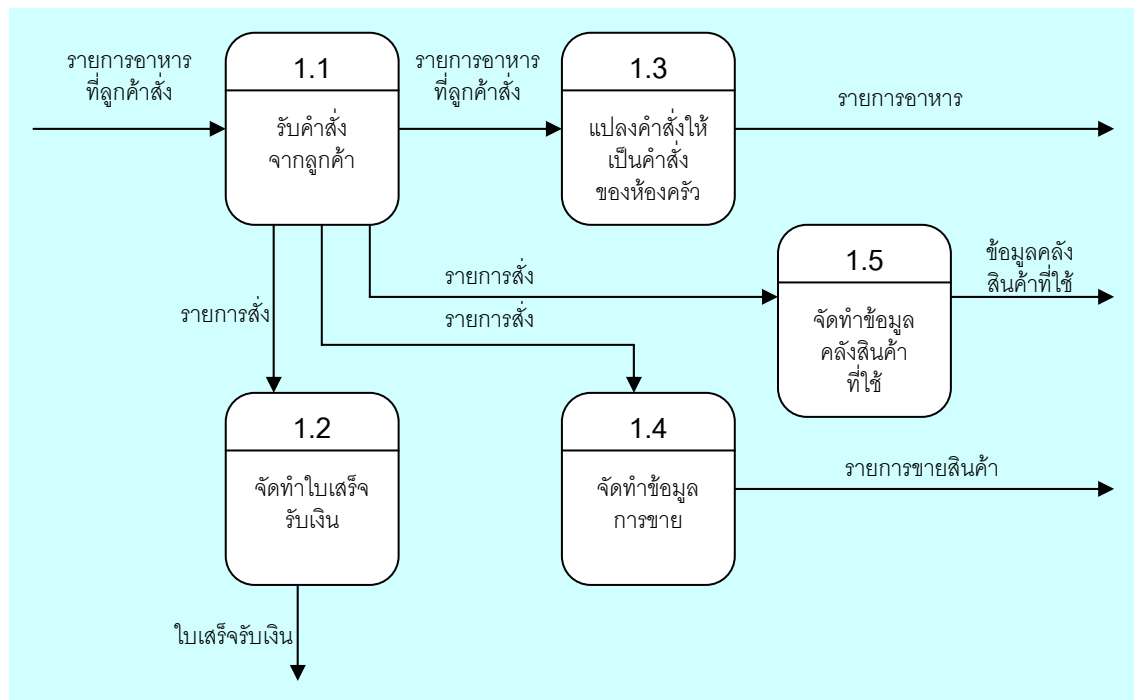
## ตัวอย่างระบบงาน : ระบบการสั่งอาหาร (Food Ordering System)



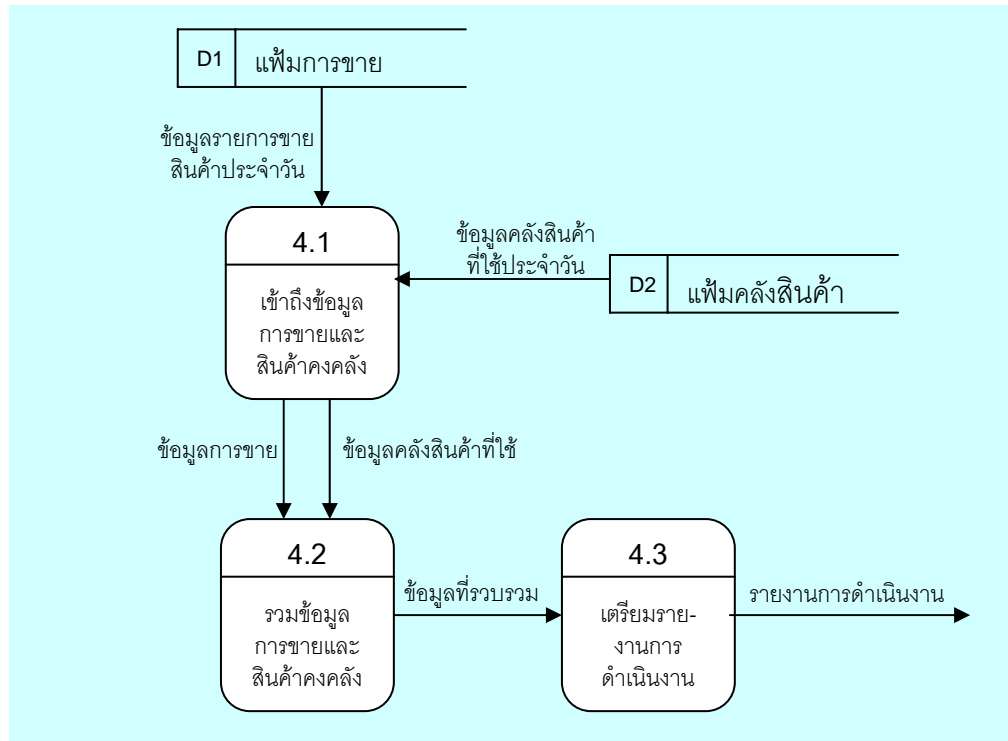
รูปที่ 4.12 Context Diagram ของระบบการสั่งอาหารของร้านอาหาร  
(ที่มา : แปลจาก Jeffrey A.Hoffer, Joey F.George, Joseph S.Valacich, 1999: 284)



รูปที่ 4.13 แผนภาพกระแสข้อมูลระดับ Diagram 1 ของระบบการสั่งอาหาร  
(ที่มา : แปลจาก Jeffrey A.Hoffer, Joey F.George, Joseph S.Valacich, 1999: 285)

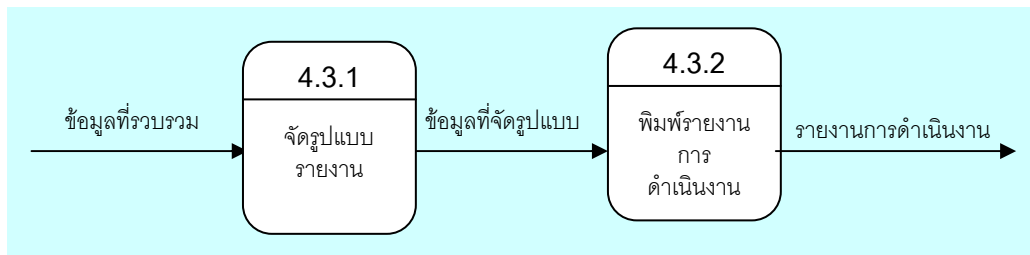


รูปที่ 4.14 แผนภาพกระแสข้อมูล Child Diagram (Level-1) ของ Process รับและส่งรายการอาหารที่ลูกค้าสั่ง  
(ที่มา : แปลจาก Jeffrey A.Hoffer, Joey F.George, Joseph S.Valacich, 1999: 289)



รูปที่ 4.15 แผนภาพกระแสข้อมูล Child Diagram (Level-1) ของ Process การสร้างรายงานการดำเนินการ  
(ที่มา : แปลจาก Jeffrey A.Hoffer, Joey F.George, Joseph S.Valacich, 1999: 289)



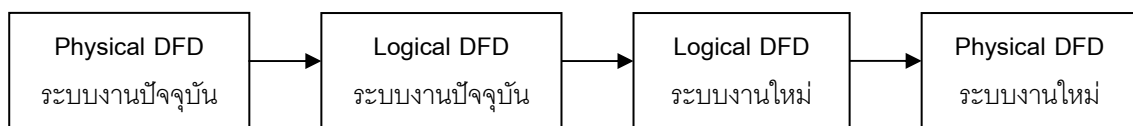


รูปที่ 4.16 แผนภาพกระแสข้อมูล Child Diagram (Level-2) ของ Process 4.3-เตรียมรายงานการดำเนินการ  
(ที่มา : แปลจาก Jeffrey A.Hoffer, Joey F.George, Joseph S.Valacich, 1999: 290)

#### 4.5.5 การพัฒนา Logical DFD ให้อยู่ในรูป Physical DFD

จากที่ได้กล่าวมาแล้วในหัวข้อประเภทของแผนภาพกระแสข้อมูลซึ่งมีอยู่ด้วยกัน 2 แบบ คือ แผนภาพกระแสข้อมูลแบบตรรกะ (Logical Data Flow Diagram) และแผนภาพกระแสข้อมูลแบบกายภาพ (Physical Data Flow Diagram) ซึ่งจากหลักการพัฒนาแผนภาพกระแสข้อมูลนั้นมีแนวคิดดังนี้

**แนวคิด :** ระบบใหม่จะถูกพัฒนาขึ้นมาจากการวิเคราะห์ระบบปัจจุบันแบบกายภาพ (Physical DFD) โดย ข้อความที่บรรจุอยู่ใน Process อาจมีชื่อบุคคลหรือตำแหน่งหรือชื่อระบบคอมพิวเตอร์ที่ทำการประมวลผลการทำงานนั้น ส่วนข้อความที่อยู่ใน Data Flow หรือ Data store อาจเป็นชื่อของสื่อทางกายภาพที่เก็บข้อมูล เช่น ชื่อแฟ้มเก็บเอกสาร , แฟ้มข้อมูลในคอมพิวเตอร์, แบบฟอร์มทางธุรกิจ หรือ เทปแม่เหล็กที่เก็บข้อมูล เป็นต้น จากนั้นเขียนแผนภาพระดับงานเดิมให้อยู่ในรูปแบบตรรกะ (Logical DFD ของระบบปัจจุบัน) โดยจะลดขั้นตอนที่เป็นการทำงานแบบกายภาพแต่จะแสดงแค่การทำงานที่ระบบเป็นผู้จัดการให้ โดยไม่สนใจรูปแบบการทำงานจริงว่าเป็นอย่างไร จากนั้นเขียนเป็น DFD ระบบงานใหม่แบบตรรกะ (Logical DFD ระบบใหม่) ซึ่งจะคล้ายกับการเขียน DFDแบบตรรกะของระบบงานเดิม เพียงแต่มีการเพิ่มฟังก์ชันการทำงานที่เพิ่มขึ้นมาหรือเปลี่ยนรูปแบบการทำงานใหม่หรือยกเลิกการทำงานแบบเก่าที่ไม่เหมาะสมออกไป เมื่อได้แผนภาพระบบงานใหม่แบบตรรกะแล้ว จึงทำการเขียนแผนภาพระบบงานใหม่แบบกายภาพ (Physical DFD ระบบงานใหม่) เพื่อแสดงการทำงานที่แท้จริงของระบบงานใหม่ ซึ่งจะสะท้อนให้เห็นการทำงานว่าในส่วนใดเป็นการทำงานที่ทำโดยคอมพิวเตอร์และส่วนใดทำงานโดยใช้คนทำ และในที่สุดระบบใหม่จะนำไปเขียนโปรแกรมได้ควรจะเป็นอย่างไร



รูปที่ 4.17 การพัฒนาระบบงานเดิมเป็นระบบงานใหม่ โดยการเขียนแผนภาพกระแสข้อมูลแบบต่าง ๆ

#### การเขียน DFDระบบงานเดิมแบบกายภาพ (Current physical DFD)

- ข้อความที่บรรจุอยู่ใน Process อาจมีชื่อบุคคลหรือตำแหน่งหรือชื่อระบบคอมพิวเตอร์ที่ทำการประมวลผลการทำงานนั้น
- ข้อความที่อยู่ใน Data Flow หรือ Data store อาจเป็นชื่อของสื่อทางกายภาพที่เก็บข้อมูล เช่น ชื่อแฟ้มเก็บเอกสาร, แฟ้มข้อมูลในคอมพิวเตอร์, แบบฟอร์มทางธุรกิจ หรือ เทปแม่เหล็กที่เก็บข้อมูล เป็นต้น

#### การเขียน DFDระบบงานเดิมแบบตรรกะ (Current logical DFD)

- จะลดขั้นตอนที่เป็นการทำงานแบบกายภาพ แต่จะแสดงแค่การทำงานที่ระบบเป็นผู้จัดการให้ โดยไม่สนใจรูปแบบการทำงานจริงว่าเป็นอย่างไร

#### การเขียน DFD ของระบบงานใหม่แบบตรรกะ (New logical DFD)

- จะคล้ายกับการเขียน DFD แบบตรรกะของระบบงานเดิม เพียงแต่มีการเพิ่มฟังก์ชันการทำงานที่เพิ่มขึ้นมาหรือเปลี่ยนรูปแบบการทำงานใหม่หรือยกเลิกการทำงานแบบเก่าที่ไม่เหมาะสมออกไป

#### การเขียน DFD ของระบบงานใหม่แบบกายภาพ (New physical DFD)

- เป็นการแสดงการทำงานที่แท้จริงของระบบงานใหม่ ซึ่งจะสะท้อนให้เห็นการทำงานว่าในส่วนตัวเป็นการทำงานที่ทำโดยคอมพิวเตอร์และส่วนใดทำงานโดยใช้คนทำ

#### การพัฒนา Logical Data Flow Diagram

ในการเขียนแผนภาพกระแสข้อมูลแบบตรรกะ (Logical) นั้นเริ่มจากการสร้าง Logical Data Flow Diagram ของระบบปัจจุบันก่อน ซึ่งจะช่วยให้ง่ายต่อการติดต่อกับผู้ใช้ระบบ เพราะทำให้เห็นถึงกิจกรรมการทำงานต่างๆ ของธุรกิจ ทำให้ผู้ใช้คุ้นเคยกับกิจกรรมที่สำคัญๆ และอาจได้ข้อมูลหรือความต้องการในแต่ละกิจกรรมเพิ่มเติมได้ นอกจากนี้ทำให้นักวิเคราะห์ระบบเข้าใจธุรกิจที่ตนเองกำลังศึกษาอยู่ จับประเด็นได้ว่าทำไมต้องมีกิจกรรมเหล่านั้น และรู้ว่ามีผลลัพธ์อะไรเกิดขึ้นบ้างจากการทำงาน

Logical Data Flow Diagram จะช่วยในการสร้างระบบให้ดีขึ้น โดยการปรับแต่งกระบวนการหรือกิจกรรมที่มีความซ้ำซ้อน หรือไม่จำเป็นออกไปจากระบบปัจจุบัน นำไปสู่กระบวนการสร้าง Logical Data Flow Diagram ของระบบใหม่

#### ความต้องการของระบบใหม่

ความต้องการระบบงานใหม่ เป็นการกำหนดแนวทางในการพัฒนาระบบงานให้มี ประสิทธิภาพมากยิ่งขึ้น โดยมีเป้าหมายคือ

1. การกำหนดแนวทางของระบบใหม่ที่จะนำมาใช้ในอนาคต
2. การสร้างกฎเกณฑ์หรือมาตรการที่จะใช้ในการตรวจสอบและประเมิน ประสิทธิภาพ ของระบบใหม่

การกำหนดความต้องการระบบงานใหม่ ต้องทำความเข้าใจกับระบบงานเดิมให้ถ่องแท้ สาเหตุที่ต้องทำความเข้าใจระบบเดิมที่ใช้อยู่ คือ

- เพื่อให้เข้าใจในหน้าที่และความรับผิดชอบของแต่ละบุคคลที่เกี่ยวข้อง
- ลักษณะงานบางอย่างมีความคล้ายคลึงกัน
- เพื่อที่จะทำความเข้าใจในลักษณะการแจกจ่ายงานในองค์กรนั้นๆ
- เพื่อที่จะแสดงให้เห็นถึงลักษณะการจัดเก็บข้อมูลที่มีความซ้ำซ้อนในระบบปัจจุบัน
- เพื่อที่จะใช้ในการตัดสินใจว่าควรจะต้องระบบเก่าไว้ โดยที่มีการอบรมผู้ใช้ เพิ่มเติม หรือให้ความรู้แก่ผู้ใช้ในเรื่องงานที่เขากำลังทำอยู่
- เพื่อที่จะค้นหาระบบการควบคุมการทำงานในระบบปัจจุบัน

### แนวทางในการกำหนดความต้องการระบบงานใหม่

การกำหนดความต้องการของระบบใหม่ เป็นการเริ่มต้นของการเตรียมการที่จะทำการออกแบบระบบใหม่ ในการกำหนดขั้นตอนของระบบใหม่ สามารถแบ่งออกได้เป็น 3 ระดับขั้นคือ

1. ขั้นตอนหลักของระบบ เป็นงานที่จะต้องทำในระบบใหม่
2. ขั้นตอนหรืองานที่เกิดขึ้นเพิ่มเติมระหว่างกาหนดความต้องการขั้นตอนหลัก ของระบบ
3. ขั้นตอนที่มีหรือไม่มีก็ได้ในระบบใหม่

สิ่งที่นักวิเคราะห์ระบบควรจะต้องให้ความสนใจในการกำหนดความต้องการของระบบ คือ

- สิ่งเข้าสู่ระบบ (Input)
- สิ่งที่ออกจากระบบ (Output)
- กระบวนการทำงาน (Operation)
- ทรัพยากร (Resource)
- มาตรการควบคุมการทำงานในแต่ละระบบงานและในทางบัญชี

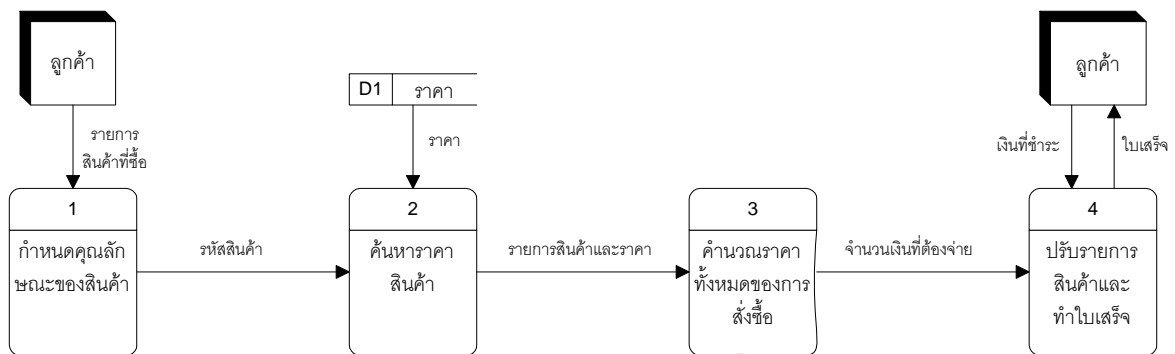
#### 4.5.2 การพัฒนา Physical Data Flow Diagram

ขั้นตอนนี้จะทำหลังจาก Logical DFD ของระบบใหม่เสร็จเรียบร้อยแล้ว ซึ่งขั้นตอนนี้จะบอกได้ว่าระบบจะถูกสร้างอย่างไร ใน Physical DFD สามารถพัฒนาโดยมีรายละเอียดต่างๆ เหล่านี้ คือ

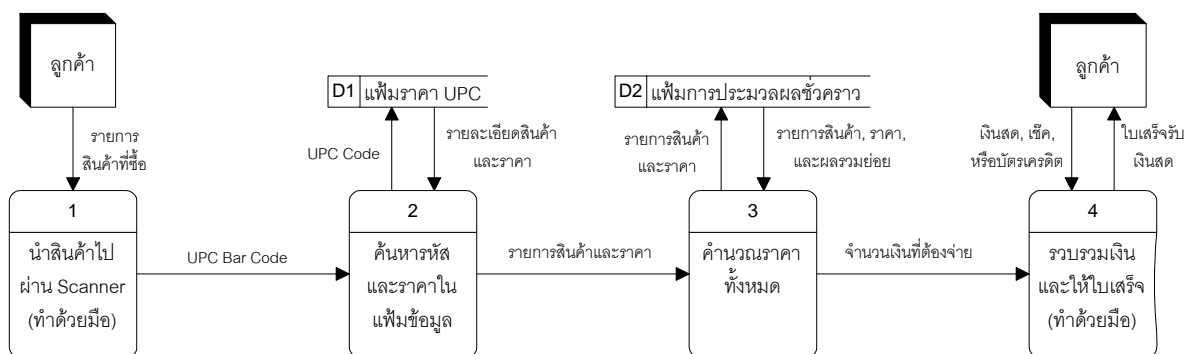
- มี Process ต่างๆ ซึ่งต้องแยกให้ชัดเจนว่า Process ไหนเป็น Process ที่ทำด้วยมือและ Process ไหนที่จะทำได้โดยอัตโนมัติ
- ในการเขียน DFDระบบงานเดิมแบบกายภาพ(Current physical DFD) ข้อความที่บรรจุอยู่ใน Process อาจมีชื่อบุคคลหรือตำแหน่งหรือชื่อระบบคอมพิวเตอร์ที่ทำการประมวลผลการดำเนินงานนั้น
- ข้อความที่อยู่ใน Data Flow หรือ Data store อาจเป็นชื่อของสื่อทางกายภาพที่เก็บข้อมูล เช่น ชื่อแฟ้มเก็บเอกสาร , แฟ้มข้อมูลในคอมพิวเตอร์, แบบฟอร์มทางธุรกิจ หรือ เทปแม่เหล็กที่เก็บข้อมูล เป็นต้น
- มี Process เกี่ยวกับการจัดลำดับเพื่อเปลี่ยนลำดับของเรคคอร์ด
- ที่เกี่ยวกับการเพิ่ม การลบ การเปลี่ยนแปลง และการปรับปรุงเรคคอร์ด

- ระบุแหล่งที่เก็บข้อมูลชั่วคราว (Temporary Data Stores)
- ระบุชื่อที่แท้จริงของแฟ้มข้อมูลหรือเอกสารที่ได้ออกมา
- มีการเพิ่มส่วนของการควบคุมลงไป เพื่อให้แน่ใจว่า Process ต่างๆ ทำงานได้อย่างถูกต้อง เช่น การปรับปรุงข้อมูลเข้า, มีสถานะฟ้องว่า “พบข้อมูล” เมื่อกำลังเข้าถึงแฟ้มข้อมูลหรือฐานข้อมูล
- บรรยายแต่ละ Process ให้ละเอียดมากกว่า Logical DFD

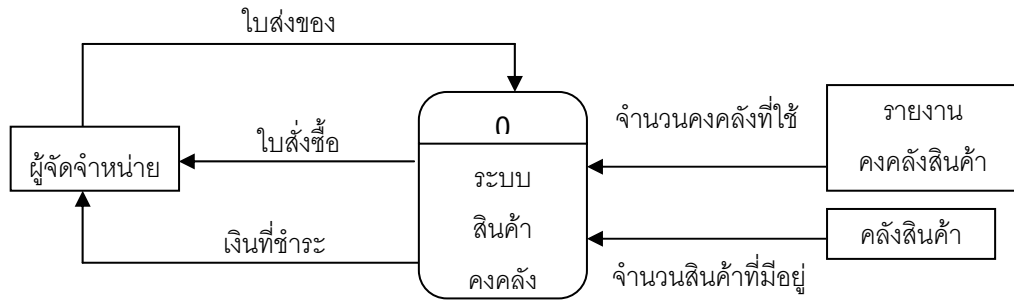
### Logical Data Flow Diagram



### Physical Data Flow Diagram



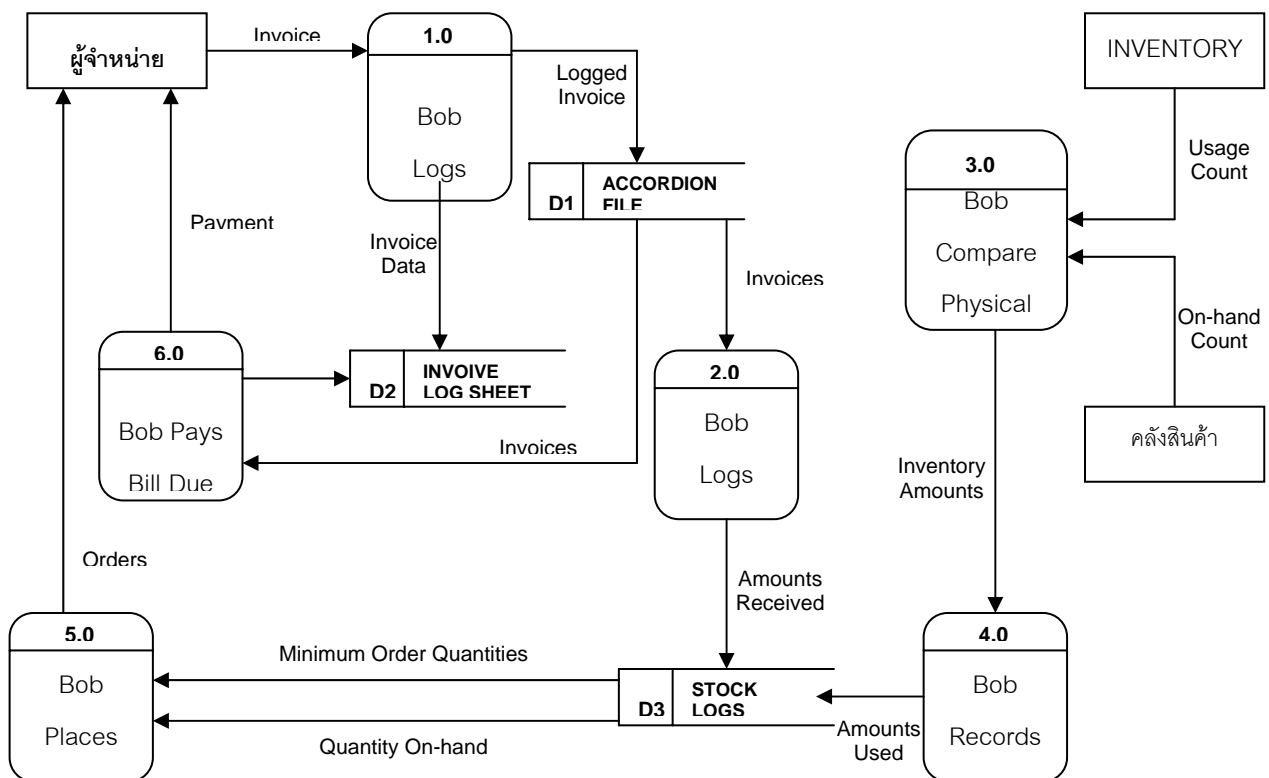
รูปที่ 4.15 ตัวอย่างการเขียน Logical DFD และ Physical DFD ของพนักงานเก็บเงินของร้านขายของชำ



Context diagram of Hoosier Burger's Manual Inventory System

จาก Context Diagram ของระบบงานเดิมแบบกายภาพ แสดงแหล่งกำเนิดของข้อมูลนอกระบบ 3 แหล่ง คือ ผู้ผลิต (Suppliers) รายงานคลังสินค้าจากระบบการขายสินค้า (Food ordering system inventory report) และ สินค้าที่เหลือในคลัง (Stock-on-hand) โดยที่ผู้ผลิตจะให้ใบส่งสินค้า (Invoice) เป็น Input เข้ามาและระบบก็จะให้ เงินที่ชำระ (Payment) และรายการสั่งซื้อ (Order) เป็น Output กลับไปให้ผู้ผลิต ส่วนรายงานคงคลังสินค้าและสินค้าที่เหลือในคลังเป็น Input ที่ได้จากระบบ

จาก Context diagram ข้างบนสามารถเขียน DFD Diagram 0 ของระบบงานเดิมแบบกายภาพ ได้ดังนี้



จากภาพเป็น DFD Diagram 0 ของ Hoosier Inventory system มี Process 6 Process ซึ่งเป็นกิจกรรมหลักที่ได้จากการวิเคราะห์ตามขั้นตอนการทำงานที่กล่าวไปแล้ว โดยมีรายละเอียดคือ Bob จะรับ Invoice จาก Supplier และบันทึกการรับสินค้าลงใน Invoice log sheet และเพิ่มข้อมูลคลังสินค้า โดย Bob จะบันทึกจำนวนสินค้าที่รับเข้ามาไว้ใน stock log ซึ่งจะเป็นแบบฟอร์มที่เป็นกระดาษที่ติดอยู่ในตำแหน่งใกล้กับสินค้านั้นเก็บอยู่ในคลัง ตัวอย่างของ stock log มีดังนี้

