บทที่ 4

การออกแบบและพัฒนาโปรแกรม

<u>สรุปขั้นตอนการทำการวางแผนกลยุทธ์สารสนเทศ</u>

จากขั้นตอนต่าง ๆ ของการวางแผนกลยุทธ์สารสนเทศ ที่ได้กล่าวมาแล้วในบทก่อนหน้านี้ สามารถสรุปเป็นขั้นตอนที่สำคัญได้ดังนี้

1 เก็บรวบรวมข้อมูลพื้นฐานที่สำคัญ ได้แก่ โครงสร้างองค์กร เป้าหมาย ปัญหา หน้าที่ และเอนทิตี (Entity) ขององค์กร

2 ระบุความสัมพันธ์ต่าง ๆ ของข้อมูลพื้นฐานที่ได้ในหัวข้อ 4.1 ในรูปของเมตริกซ์เป็นคู่ลำดับได้แก่

2.1 โครงสร้างขององค์กร และ เป้าหมาย

2.2 โครงสร้างขององค์กร และ ปัญหา

2.3 หน้าที่ และ โครงสร้างขององค์กร

2.4 หน้าที่ และ เป้าหมาย

2.5 หน้าที่ และ ปัญหา

2.6 เอนทิตี และ โครงสร้างขององค์กร

2.7 เอนทิตี และ เป้าหมาย

2.8 เอนทิตี และ ปัญหา

3 กำหนดแบบจำลองธุรกิจ คือ การระบุความสัมพันธ์ระหว่างหน้าที่และเอนทิตี โดยระบุว่าหน้าที่ใดเป็นผู้สร้าง หรือ ปรับปรุง หรือ ลบ หรือ อ่านเอนทิตี หรือ การจัดเรียงเอนทิตี โดยให้ข้อมูลที่ถูกสร้าง (C) เรียงจากบนมาล่าง จากซ้ายไปขวา หรือที่เรียกว่ากฎ “Northwest”

4 การคำนวณค่าความสัมพันธ์ระหว่างเอนทิตี

5 กำหนดกลุ่มความสัมพันธ์ โดยเลือกหน้าที่และเอนทิตี มีความสัมพันธ์ใกล้ชิดกันมาอยู่ในระบบย่อยเดียวกัน

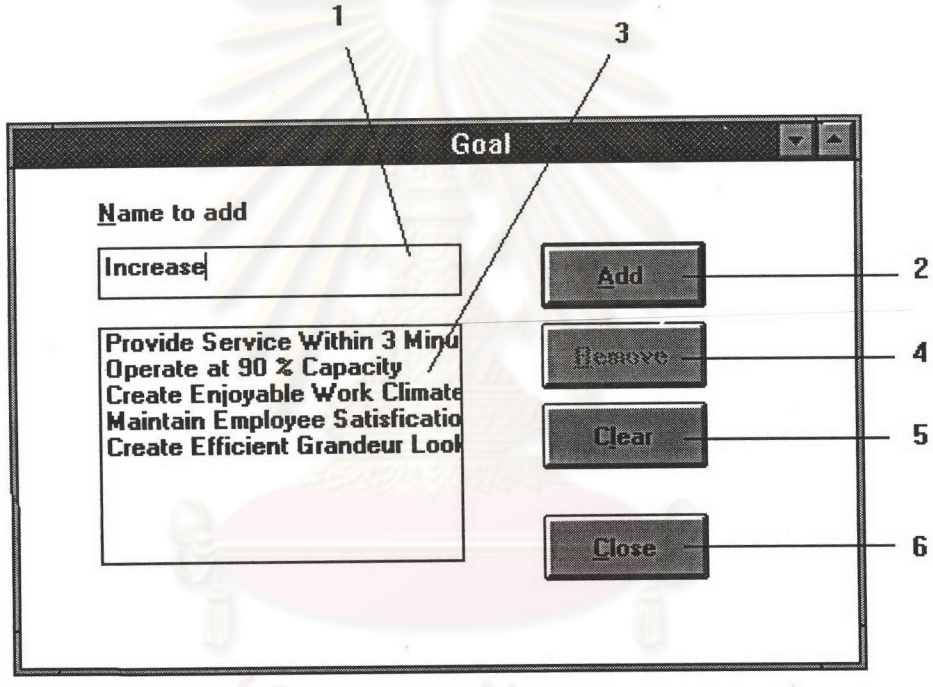
6 สร้างความสัมพันธ์ระหว่างระบบย่อยเหล่านั้น โดยระบุว่าระบบใดเป็นระบบที่สร้างข้อมูลเพื่อให้ระบบอื่นนำไปประมวลผลต่อ

การออกแบบโปรแกรม

ขั้นตอนการออกแบบโปรแกรม จะสอดคล้องไปตามขั้นตอนการวางแผนกลยุทธ์สารสนเทศ ที่สรุปมาแล้วในหัวข้อ 1 ดังนั้นจะเริ่มจากการออกแบบหน้าจอการรับข้อมูล ซึ่งแบ่งไปตามหัวข้อดังนี้

1 การออกแบบหน้าจอส่วนรับข้อมูลพื้นฐาน

ข้อมูลพื้นฐานจะประกอบด้วย โครงสร้างองค์กร เป้าหมาย ปัญหา หน้าที่ และ เอนทิตี ซึ่งจะมีหน้าจอการรับข้อมูล ดังรูปที่ 4.1



รูปที่ 4.1 รูปแสดง ตัวอย่างหน้าจอรับข้อมูลพื้นฐาน

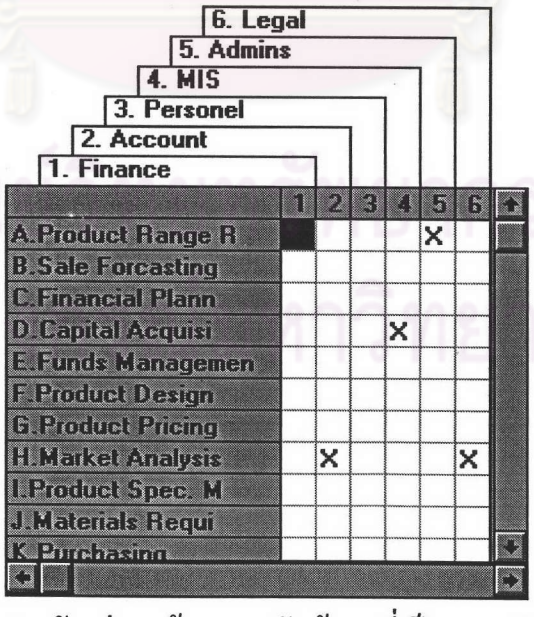
หมายเลข 1 เป็นบริเวณที่ป้อนข้อมูล
หมายเลข 2 เป็นปุ่มยอมรับการเพิ่มรายการ
หมายเลข 3 เป็นบริเวณที่แสดงข้อมูลที่ใส่ไปแล้ว
หมายเลข 4 เป็นปุ่มลบรายการที่เลือก
หมายเลข 5 เป็นปุ่มลบรายการทั้งหมด
หมายเลข 6 เป็นปุ่มปิดการรับข้อมูล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2 การออกแบบหน้าจอส่วนรับความสัมพันธ์ระหว่างข้อมูลพื้นฐาน

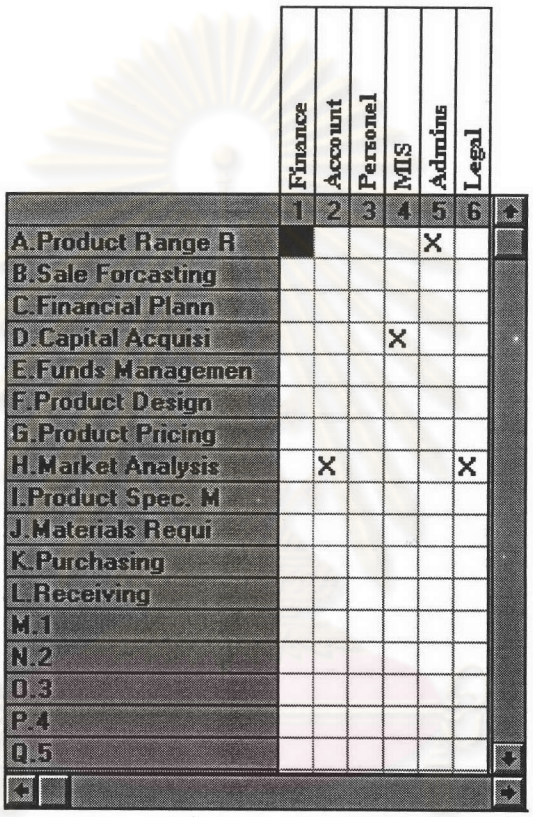
การระบุความสัมพันธ์ระหว่างข้อมูลพื้นฐานจะมีอยู่หลายแบบ แต่วิธีการระบุความสัมพันธ์ก็จะคล้าย ๆ กัน จะมีลักษณะหน้าจอการรับข้อมูล ดังรูปที่ 4.2 โดยโปรแกรมเคส จะดึงรายการข้อมูลพื้นฐานมาสร้างเป็นเมตริกซ์ และการระบุความสัมพันธ์ทำได้โดยใช้เมาส์ ซึ่งสัญญลักษณ์ที่แสดงความสัมพันธ์ อาจเป็นตัวอักษร “X” ,”I” หรือ “D” ขึ้นอยู่กับประเภทความสัมพันธ์นั้น โดย “X” หมายถึง มีความสัมพันธ์ “I” หมายถึง มีความสัมพันธ์โดยอ้อม และ “D” หมายถึง มีความสัมพันธ์โดยตรง

ประเภทของความสัมพันธ์	สัญญลักษณ์ที่ใช้
โครงสร้างขององค์กร และ เป้าหมาย	X
โครงสร้างขององค์กร และ ปัญหา	X
หน้าที่ และ โครงสร้างขององค์กร	X
หน้าที่ และ เป้าหมาย	I,D
หน้าที่ และ ปัญหา	I,D
เอนทิตี และ โครงสร้างขององค์กร	X
เอนทิตี และ เป้าหมาย	I,D
เอนทิตี และ ปัญหา	I,D



รูปที่ 4.2 รูปแสดงตัวอย่างหน้าจอภาพรับข้อมูลที่เป็นความสัมพันธ์ แบบที่ 1

แต่เนื่องจากการระบุความสัมพันธ์ตามหน้าจอ รูปที่ 4.2 จะรับข้อมูลได้น้อยเพราะจำนวนรายการที่อยู่ในแนวคอลัมน์ยิ่งมากจะทำให้จำนวนแถวของการแสดงข้อมูลน้อยลง จึงออกแบบการรับข้อมูลอีกแบบตามรูปที่ 4.3 โดยการเพิ่มรายการตามแนวคอลัมน์ จะไม่มีผลทำให้จำนวนแถวในการแสดงผลน้อยลง แต่มีข้อเสียคือ การอ่านรายการข้อมูลตามคอลัมน์ ทำได้ลำบาก

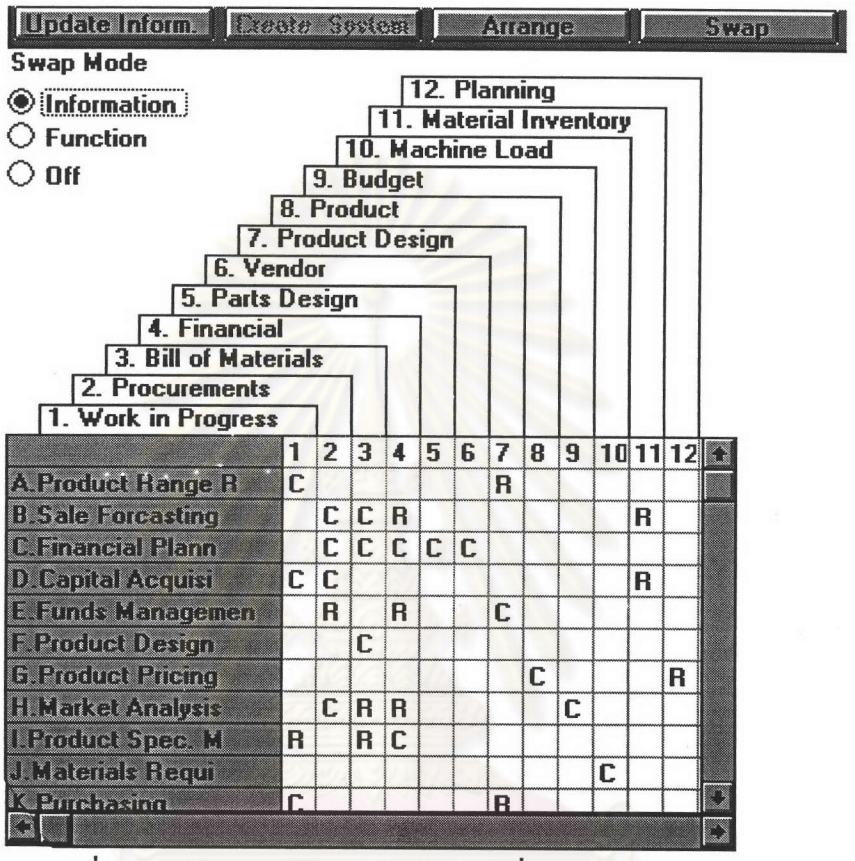


รูปที่ 4.3 รูปแสดงตัวอย่างหน้าจอภาพรับข้อมูลที่เป็นความสัมพันธ์ แบบที่ 2

3 การออกแบบหน้าจอเพื่อสร้างแบบจำลองธุรกิจ

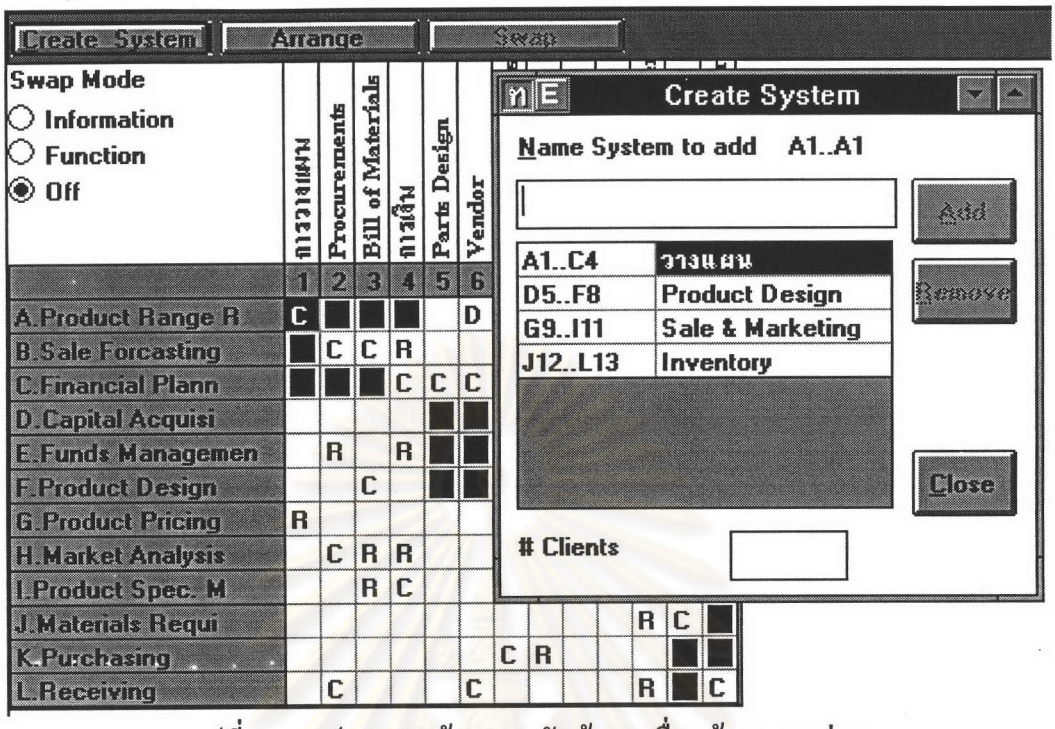
การระบุความสัมพันธ์ในแบบจำลองธุรกิจ จะใช้เมตริกซ์เหมือนข้อ 2 การระบุความสัมพันธ์ทำได้โดยใช้เมาส์ และสัญญลักษณ์ที่แสดงความสัมพันธ์ จะเป็นตัวอักษร “C”, ”D”, ”U” หรือ “R” ขึ้นอยู่กับประเภทความสัมพันธ์นั้น โดย “C” หมายถึง การสร้างข้อมูล “D” หมายถึง การลบข้อมูล “U” หมายถึง การแก้ไขข้อมูล และ “R” หมายถึง การอ่านข้อมูล ถ้ามีความสัมพันธ์หลายแบบ ให้ระบุเพียงอย่างเดียว โดยให้เลือกตามลำดับความสำคัญดังนี้ C,D,U และ R นอกจากระบุความสัมพันธ์แล้วโปรแกรมยังสามารถสลับข้อมูล โดยใช้ปุ่ม “Swap” และถ้าต้องการให้โปรแกรมเรียงเอนทิตีให้เป็นไปตามกฎ “Northwest” โดยใช้ปุ่ม “Arrange” จะได้ความสัมพันธ์ที่เป็นการสร้างข้อมูล (C) เรียงจากบนลงล่าง จากซ้ายไปขวา หน้าจอส่วนนี้จะได้ดังรูปที่ 4.4 และนอกจากนี้ยังต้องสามารถสร้าง

ระบบย่อย ๆ ได้ โดยใช้เมาส์เลือกบริเวณที่ต้องการแล้วใช้ปุ่ม “Create” จะได้หน้าจอภาพที่ทำหน้ารับชื่อระบบ หน้าจอส่วนนี้จะได้ดังรูปที่ 4.5



รูปที่ 4.4 รูปแสดงหน้าจอการรับข้อมูลเพื่อสร้างแบบจำลองธุรกิจ

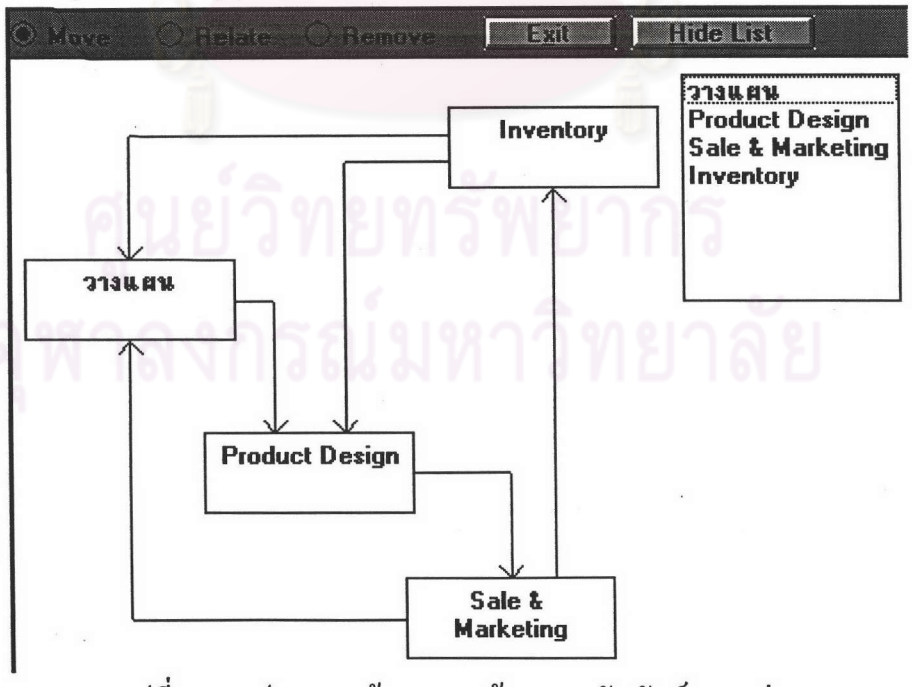
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.5 รูปแสดงหน้าจอการรับข้อมูลเพื่อสร้างระบบย่อย

4 การออกแบบหน้าจอเพื่อสร้างความสัมพันธ์ระบบย่อย

การระบุความสัมพันธ์ระหว่างระบบย่อย ใช้เส้นโยงจากระบบหนึ่งไปยังระบบหนึ่ง หน้าจอการรับข้อมูลจะเป็นดังรูปที่ 4.6



รูปที่ 4.6 รูปแสดงหน้าจอการสร้างความสัมพันธ์ระบบย่อย

5 การออกแบบหน้าจอเพื่อแสดงค่าความสัมพันธ์ระหว่างเอนทิตี

ค่าความสัมพันธ์ระหว่างเอนทิตี สามารถแสดงได้ในลักษณะของเมตริกซ์ ซึ่งแสดงได้ตามรูปที่ 4.7 วิธีการคำนวณดูได้จากบทที่ 3 หัวข้อ 3.1.1 ประโยชน์ที่ได้รับจากการใช้เคสตรงจุดนี้ก็คือ ผู้ออกแบบระบบตามวิธีการวางแผนกลยุทธ์สารสนเทศ ไม่ต้องนับความสัมพันธ์ระหว่างเอนทิตีซึ่งต้องใช้เวลามาก และมีโอกาสผิดพลาดได้ โดยที่โปรแกรมเคสนี้ จะดึงความสัมพันธ์จากแบบจำลองธุรกิจขึ้นมาคำนวณเอง ทำให้สะดวกต่อการทำงานมาก

		E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9
Work in Progress	E 1	0	0.25	0.25	0.25	0.00	0.00	0.50	0.00	0.00
Procurements	E 2	0.17	0	0.50	0.67	0.17	0.33	0.17	0.00	0.17
Bill of Materials	E 3	0.20	0.60	0	0.80	0.20	0.20	0.00	0.00	0.20
Financial	E 4	0.20	0.80	0.80	0	0.20	0.20	0.20	0.00	0.20
Parts Design	E 5	0.00	1.00	1.00	1.00	0	1.00	0.00	0.00	0.00
Vendor	E 6	0.00	1.00	0.50	0.50	0.50	0	0.00	0.00	0.00
Product Design	E 7	0.67	0.33	0.00	0.33	0.00	0.00	0	0.00	0.00
Product	E 8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0.00
Budget	E 9	0.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0

รูปที่ 4.7 รูปแสดงหน้าจอแสดงค่าความสัมพันธ์ระหว่างเอนทิตี

6 การออกแบบรายงาน

รายงานสำหรับโปรแกรม CASE for ISP ก็คือ ผลลัพธ์ที่หน้าจอภาพต่าง ๆ จะมีรายงานดังนี้

6.1 แสดงรายละเอียดกลุ่มข้อมูลพื้นฐาน 6 รายงาน ตัวอย่างรูปที่ 4.8

Chula Co.,Ltd. Function Page 1

Project Date: dd/mm/yyyy dd/mm/yy hh:mm

Function List

1. Product Range
2. Sale Forcasting
3. Financial Plan

.....

รูปที่ 4.8 รูปแสดง ตัวอย่างรายงานข้อมูลพื้นฐาน

6.2 แสดงความสัมพันธ์ระหว่างข้อมูลพื้นฐาน ตัวอย่างรูปที่ 4.9

Chula Co.,Ltd. Business Function Analysis Page 1

Project Date: dd/mm/yyyy dd/mm/yy hh:mm

	1 Finance	2 Account	3 Personel	4 MIS	5 Admins	6 Legal
A.Product Range R					X	
B.Sale Forcasting						
C.Financial Plann						
D.Capital Acquisi				X		
E.Funds Managemen						
F.Product Design						
G.Product Pricing						
H.Market Analysis		X				X
I.Product Spec. M						
J.Materials Requi						
K.Purchasing						
L.Receiving						

รูปที่ 4.9 รูปตัวอย่าง แสดงความสัมพันธ์ระหว่างข้อมูลพื้นฐาน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

6.3 แสดงแบบจำลองธุรกิจ ตามรูปที่ 4.10

Chula Co.,Ltd. Business Information Model Page 1

Project Date: dd/mm/yyyy dd/mm/yy hh:mm

	Table	Reservation	Other Establishm	Guest	Order	Staff	Furnishings & Eq	Ingredient	Invoice	Menu Line Item	General Ledger	Course	Facility
	1	2	3	4	5	6	7	8	9	10	11	12	13
A.Analyze Trends													U
B.Review Local Ma		C							R				
C.Adopt Changes											D	D	
D.Process Reserva			C					D					
E.Enter Order				C									
F.Present Cuisine				D	C	C							
G.Present Bill													
H.Receive Order		D		U									
I.Schedule Order			C										
J.Prepare Courses							C		U				
K.Analyze Usage								C					
L.Forecast Requir									C				

รูปที่ 4.10 รูปแสดงแบบจำลองธุรกิจ

6.4 แสดงตารางค่าความสัมพันธ์ระหว่างเอนทิตี ตามรูปที่ 4.11

Chula Co.,Ltd. Affinity Table Page 1

Project Date: dd/mm/yyyy dd/mm/yy hh:mm

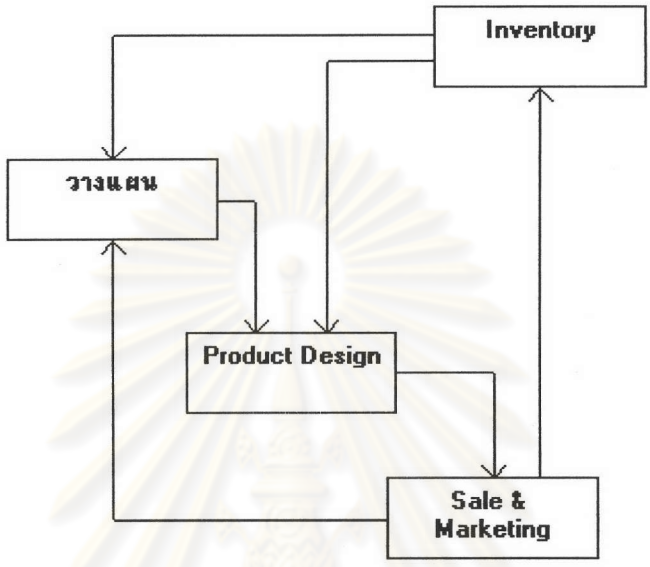
		E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9
Work in Progress	E 1	0	0.25	0.25	0.25	0.00	0.00	0.50	0.00	0.00
Procurements	E 2	0.17	0	0.50	0.67	0.17	0.33	0.17	0.00	0.17
Bill of Materials	E 3	0.20	0.60	0	0.80	0.20	0.20	0.00	0.00	0.20
Financial	E 4	0.20	0.80	0.80	0	0.20	0.20	0.20	0.00	0.20
Parts Design	E 5	0.00	1.00	1.00	1.00	0	1.00	0.00	0.00	0.00
Vendor	E 6	0.00	1.00	0.50	0.50	0.50	0	0.00	0.00	0.00
Product Design	E 7	0.67	0.33	0.00	0.33	0.00	0.00	0	0.00	0.00
Product	E 8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0.00
Budget	E 9	0.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0

รูปที่ 4.11 รูปแสดงตารางค่าความสัมพันธ์ระหว่างเอนทิตี

6.5 แสดงความสัมพันธ์ระหว่างระบบย่อย ตามรูปที่ 4.12

Chula Co.,Ltd. System Dependency Page 1

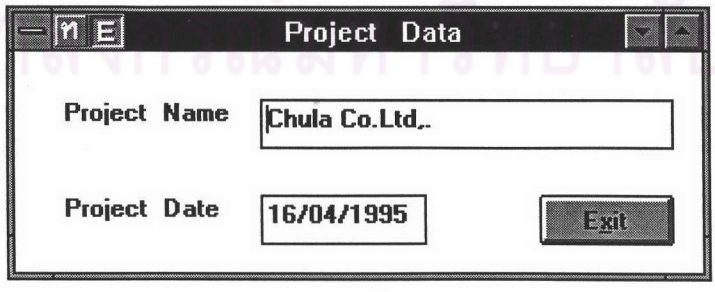
Project Date: dd/mm/yyyy dd/mm/yy hh:mm



รูปที่ 4.12 รูปแสดงความสัมพันธ์ระหว่างระบบย่อย

7 การออกแบบระบบและเมนู

เพื่อให้ผู้ใช้ระบบสามารถใช้งานเคสนี้ได้ง่าย ผู้ใช้จึงออกแบบระบบโดย ใช้เมนูเป็นหลัก เพื่อให้โปรแกรมย่อยต่าง ๆ สามารถทำงานต่อกันได้ ก่อนอื่นเมื่อเริ่มโครงการใหม่ จะต้องมีการตั้งชื่อโครงการและวันที่เริ่มโครงการดังรูป 4.13



รูปที่ 4.13 รูปแสดงหน้าจอการรับชื่อโครงการและวันที่

โดยระบบมีเมนูจะเป็นแบบเมนูดึงลง มีโครงสร้างหลักดังนี้

> "&File" **เรียกเมนูย่อยเกี่ยวกับไฟล์**

>>>> "&New Project" เริ่มโปรเจกต์ใหม่

>>>> "&Load Project" โหลดแฟ้มโปรเจกต์

>>>> "&Save Project " จัดเก็บแฟ้มโปรเจกต์

>>>> "-" เส้นแบ่งเมนู

>>>> "Print &Setup" จัดการพิมพ์

>>>> "&Print" พิมพ์รายงาน

>>>> "-" เส้นแบ่งเมนู

>>>> "E&xit" จบการทำงาน

> "&Categories" **เรียกเมนูย่อยเกี่ยวกับข้อมูลพื้นฐาน**

>>>> "Project Data" เก็บชื่อโปรเจกต์

>>>> "-" เส้นแบ่งเมนู

>>>> "Goal" เก็บข้อมูลเป้าหมาย

>>>> "Problem" เก็บข้อมูลปัญหา

>>>> "Organization" เก็บข้อมูลหน่วยองค์กร

>>>> "Information Class" เก็บข้อมูลเอนทิตี

>>>> "Function" เก็บข้อมูลหน้าที่

> "&Grid " **เรียกเมนูย่อยเกี่ยวกับเมตริกซ์**
เพื่อแสดงความสัมพันธ์คู่ต่าง ๆ

>>>> "&Goal && Problem Analysis" เรียกเมนูย่อย

>>>>>>> "Organization && &Goal"

>>>>>>> "Organization && &Problem"

>>>> "&Bussiness Function Analysis" เรียกเมนูย่อย

>>>>>>> "Function && &Organization"

>>>>>>> "Function && &Goal"

>>>>>>> "Function && &Problem"

>>>> "&Information Analysis" เรียกเมนูย่อย

>>>>>>> "Information && &Organization"

>>>>>>> "Information && &Goal"

>>>>>>> "Information && &Problem"

>>>> "Business Information &Model" สร้างแบบจำลองธุรกิจ

>>>> "-"

>>>> "&Normal Style" เลือกแบบแสดงผลของเมตริกซ์

> "&Utilities" **เรียกเมนูย่อย**

>>>> "&Draw System Dependency" สร้างความสัมพันธ์ระหว่างระบบย่อย

>>>> "Affinity &Table" แสดงค่าความสัมพันธ์ระหว่างเอนทิตี

> "&Select Print" **เรียกเมนูย่อย เลือกการพิมพ์ แบบต่าง ๆ**

>>>> "Function && &Organization"

>>>> "Function && &Goal"

>>>> "Function && &Problem"

>>>> "-"

>>>> "Information && &Organization"

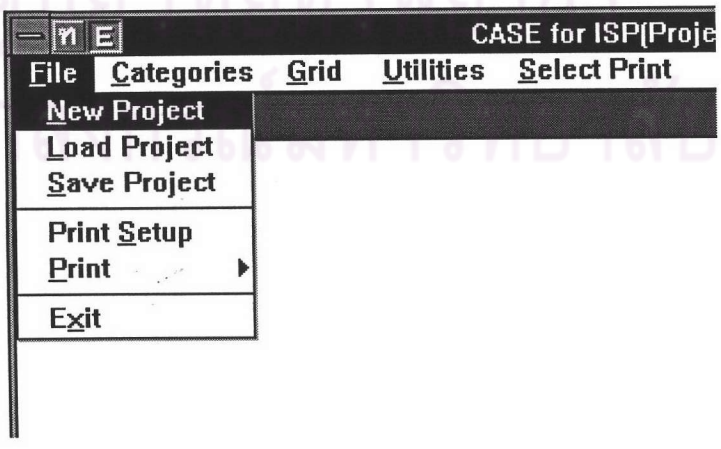
>>>> "Information && &Goal"

>>>> "Information && &Problem"

>>>> "-"

>>>> "Business Information &Model"

เมนูที่แสดงนี้ เมื่อนำมาแสดงที่หน้าจอภาพ จะได้ตามรูปที่ 4.14



รูปที่ 4.14 รูปแสดงเมนูของระบบ

I16592529

การพัฒนาโปรแกรม

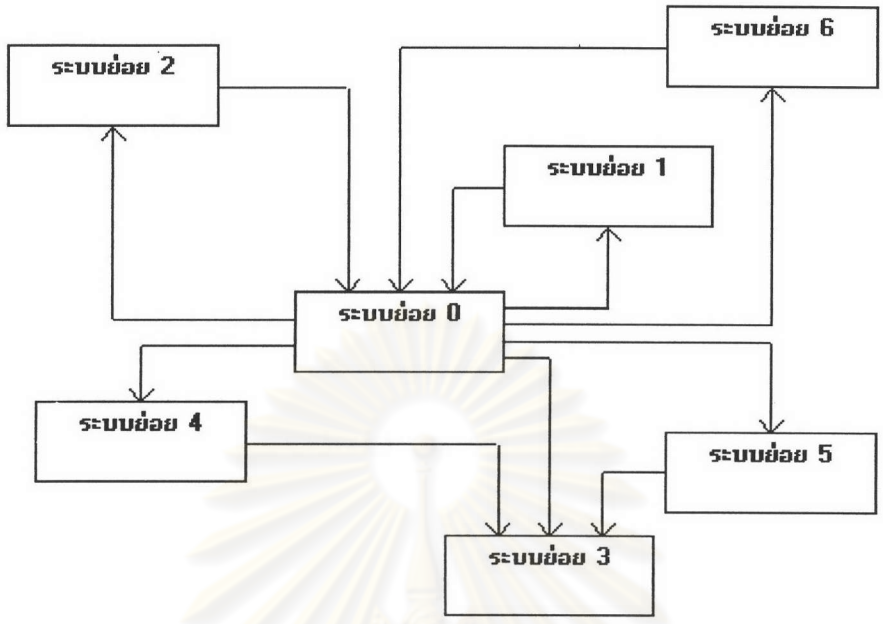
ในปัจจุบันโปรแกรมวินโดส์ได้รับความนิยม ในกลุ่มผู้ใช้งานเครื่องคอมพิวเตอร์ส่วนบุคคลอย่างแพร่หลาย และมีแนวโน้มที่จะพัฒนาให้มีประสิทธิภาพดียิ่งขึ้น และสามารถใช้งานบนเครื่องที่สถาปัตยกรรมที่ต่างออกไป เช่น Power PC เนื่องจากการใช้โปรแกรมประยุกต์ต่าง ๆ บนโปรแกรมวินโดส์ มีความง่ายและเป็นที่ยอมรับทั่วไป ดังนั้น โปรแกรม CASE for ISP จึงถูกออกแบบให้พัฒนาบนวินโดส์

เมื่อสรุปว่าการพัฒนาโปรแกรม CASE for ISP จะพัฒนาบนระบบวินโดส์ ต่อไปก็คือการเลือกภาษาที่ใช้ในการพัฒนาระบบ ปัจจุบันมีโปรแกรมภาษาหลาย ๆ ภาษาที่ถูกพัฒนาเพื่อเขียนโปรแกรมบนระบบวินโดส์มีอยู่มากมาย ตัวอย่างเช่น Visual C++, Turbo C++, PowerBuilder, Visual Basic และอื่น ๆ อีกมาก แต่เป็นที่ยอมรับกันว่าโปรแกรมภาษาที่ได้รับการยอมรับจากนักพัฒนาโปรแกรมทั่วไป มีเครื่องมือที่ช่วยในการพัฒนาโปรแกรมมากและพัฒนาโปรแกรมได้สะดวก รวดเร็ว คือ วิชวลเบสิก (Visual Basic) ซึ่งในขณะที่พัฒนาโปรแกรม โปรแกรมวิชวลเบสิกได้พัฒนาถึงเวอร์ชั่น 3.0

การเขียนโปรแกรมด้วยวิชวลเบสิก จะใช้แนวคิดในการเขียนโปรแกรมแบบใหม่ที่เรียกว่า Event-Driven Programing (การเขียนโปรแกรมโดยเหตุการณ์) นอกจากนี้ยังมีเครื่องมือช่วย (Tool Box) ในการพัฒนาระบบ เครื่องมือแต่ละชิ้นจะถูกเรียกว่าคอนโทรล (Control) ซึ่งคอนโทรล แต่ละชิ้นจะมีลักษณะเป็นวัตถุ (Object) การเขียนโปรแกรมโดยใช้วัตถุนี้จะเรียกว่าการโปรแกรมเชิงวัตถุ (Object-Oriented Programing) คอนโทรลในวิชวลเบสิกจะมีอยู่หลายชิ้น และยังมีผู้พัฒนาคอนโทรลต่าง ๆ เพื่อให้สามารถพัฒนาโปรแกรมด้วยวิชวลเบสิกได้สะดวกขึ้น

การออกแบบโครงสร้างข้อมูล และการเขียนโปรแกรม

การวาดเส้นโยงความสัมพันธ์ระหว่างระบบย่อย เพื่อให้เกิดความสวยงามในการแสดงผล ดังตัวอย่างในรูปที่ 4.15 จำเป็นจะต้องมีการคำนวณหาข้อมูล ที่จำเป็นในการวาดเส้นความสัมพันธ์



รูปที่ 4.15 รูปแสดงเส้นความสัมพันธ์ระหว่างระบบย่อย

เพื่อไม่ให้วิธีการลากเส้นความสัมพันธ์มีความยุ่งยากมากเกินไป จึงมีข้อกำหนดดังนี้

1. เส้นที่ลากออกจากระบบย่อย เป็นเส้นตรงในแนวระดับ จะลากไปทางซ้ายหรือทางขวาก็ได้
2. เส้นที่ลากเข้าหาระบบย่อย เป็นเส้นตรงในแนวดิ่ง จะมาจากด้านบนหรือด้านล่างก็ได้
3. เส้นความสัมพันธ์จะประกอบด้วย เส้นตรงไม่เกิน 2 เส้น
4. ระบบย่อยจะแทนด้วยรูปสี่เหลี่ยม

เพื่อให้เรียกชื่อระบบย่อยให้ชัดเจน จึงขอตั้งชื่อระบบย่อยที่มีความสัมพันธ์กันดังนี้ ระบบย่อยที่มีเส้นความสัมพันธ์ลากออกมาเรียกว่า ระบบย่อยที่ให้กำเนิดข้อมูล ส่วนระบบย่อยที่มีเส้นความสัมพันธ์ชี้เข้าไปเรียกว่า ระบบย่อยที่ใช้ข้อมูล โดยที่ในระบบย่อยเดียวกัน อาจมีสถานะเป็น ระบบย่อยที่ให้กำเนิดข้อมูล และระบบย่อยที่ใช้ข้อมูล ก็ได้

หลังจากมีข้อกำหนดนี้แล้ว การลากเส้นความสัมพันธ์ จะต้องมีข้อมูลเพื่อพิจารณาในการลากเส้น โดยแบ่งการพิจารณาเป็น 2 ลักษณะ

1. ข้อมูลที่ใช้เมื่อมีเส้นความสัมพันธ์ลากออกจากระบบย่อย

1.1 ข้อมูลจำนวนระบบย่อยที่ใช้ข้อมูล ที่อยู่ทางด้านซ้าย หรืออยู่ทางด้านขวาของระบบย่อยที่ให้กำเนิดข้อมูล อย่างใดอย่างหนึ่ง โดยดูจากตำแหน่งบนจอภาพ เพื่อใช้คำนวณตำแหน่ง จุดที่

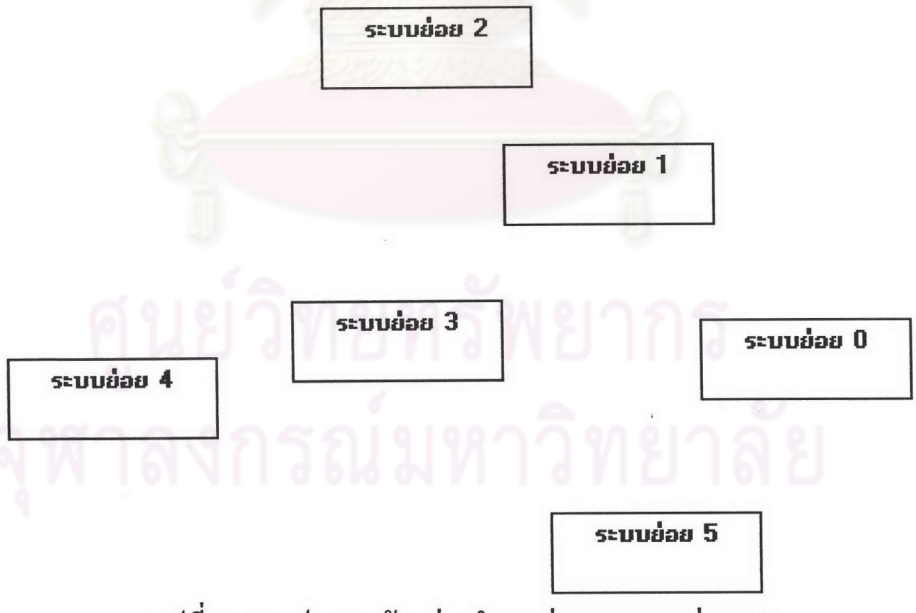
เส้นความสัมพันธ์ถูกลากออกจากระบบย่อยที่ให้กำเนิดข้อมูล เช่น ถ้าทางด้านซ้ายของระบบย่อย ที่ให้กำเนิดข้อมูล มีระบบย่อยที่นำข้อมูลไปใช้อยู่ 3 ระบบ และ ถ้าด้านในแนวดิ่งของสี่เหลี่ยมที่แทนระบบมีขนาด 400 จุด (หน่วยความยาวของด้านบนจอภาพ) วิธีการคำนวณระยะห่างระหว่างจุดคือ

ความยาวของด้านในแนวดิ่ง / (จำนวนระบบย่อยที่ใช้ข้อมูล + 1)

ดังนั้นแต่ละจุดที่เป็นจุดเริ่มต้นของเส้นความสัมพันธ์ จะมีระยะห่าง เท่ากับ 400/(3+1) = 100 จุด หมายความว่า จุดแรกที่ลากเส้นความสัมพันธ์ออกไปจะอยู่ห่างจากมุมบนของสี่เหลี่ยมลงมาด้านล่างเป็นระยะ 100 จุด และเส้นความสัมพันธ์เส้นที่สอง จะอยู่ต่ำลงมาจากมุมของสี่เหลี่ยมเป็นระยะ 200 จุด ถ้าระบบย่อยที่ใช้ข้อมูลอยู่ทางด้านขวามือ ก็มีวิธีการคำนวณอย่างเดียวกัน

1.2 ข้อมูลลำดับของระบบย่อย ที่จะลากเส้นความสัมพันธ์ออกไป เป็นการเลือกระบบย่อยที่จะต้องนำจุดต่าง ๆ ในแนวดิ่งของสี่เหลี่ยมที่คำนวณได้ในหัวข้อ 1.1 เป็นจุดที่เส้นความสัมพันธ์ถูกลากออกไปจากระบบที่ให้กำเนิดข้อมูล โดยการเรียงระบบย่อยที่ใช้ข้อมูล มีวิธีการเรียงดังนี้

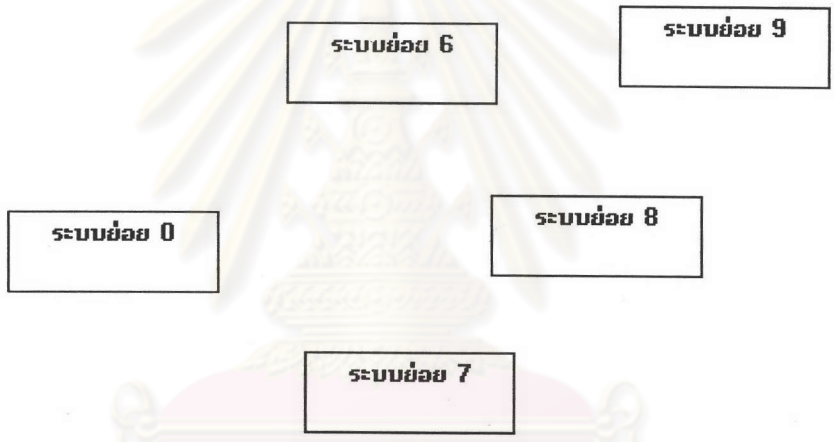
1.2.1 แบ่งระบบย่อยที่ใช้ข้อมูล เป็น 2 กลุ่ม โดยแบ่งตามตำแหน่งบนจอภาพ ถ้าระบบย่อยที่ใช้ข้อมูล อยู่สูงกว่า ระบบย่อยที่ให้กำเนิดข้อมูล จัดให้อยู่ในกลุ่มที่ 1 ถ้าอยู่ต่ำกว่า ระบบย่อยที่ให้กำเนิดข้อมูล ให้จัดอยู่ในกลุ่มที่ 2 ถ้ามีความสัมพันธ์ตามรูปที่ 4.16 (ระบบย่อย 0 เป็นระบบย่อยที่ให้กำเนิดข้อมูล) ระบบย่อยที่ 1,2,3 จะอยู่กลุ่มที่ 1 ส่วนระบบย่อยที่ 4,5 จะอยู่กลุ่มที่ 2



รูปที่ 4.16 รูปแสดงตัวอย่างตำแหน่งของระบบย่อย (1)

1.2.2 ลำดับระบบย่อยที่ใช้ข้อมูล เพื่อเรียงการลากเส้นความสัมพันธ์จากบนลงล่าง โดยจะเรียงในกลุ่มที่ 1 ก่อน (กลุ่มระบบย่อยที่ใช้ข้อมูลอยู่สูงกว่าระบบย่อยที่ให้กำเนิดข้อมูล) โดย

หลักการเรียงก็คือ ระบบย่อยที่ใช้ข้อมูล อยู่ใกล้กับระบบย่อยที่ให้กำเนิดข้อมูลมากที่สุด โดยคำนึงถึงความใกล้ในแนวระดับเป็นเกณฑ์ จัดให้เป็นลำดับที่ 1 ส่วนระบบย่อยอื่นที่ห่างออกไปก็จัดเป็นลำดับที่ 2,3,4 ... N ไปเรื่อย ๆ (N แทนจำนวนระบบย่อยที่ใช้ข้อมูล ในกลุ่มที่ 1) ในกลุ่มที่ 2 ก็เช่นกัน (กลุ่มระบบย่อยที่ใช้ข้อมูลอยู่ต่ำกว่าระบบย่อยที่ให้กำเนิดข้อมูล) คือ ระบบย่อยที่ใช้ข้อมูล อยู่ใกล้กับระบบย่อยที่ให้กำเนิดข้อมูลมากที่สุด โดยคำนึงถึงความใกล้ในแนวระดับเป็นเกณฑ์ จัดให้เป็นลำดับที่ N+1 ส่วนระบบย่อยอื่นที่ ห่างออกไปก็จัดเป็นลำดับที่ N+2,N+3,N+4 ... ดูตัวอย่างจากรูป 4.16 จะลำดับระบบย่อยได้ดังนี้ คือ 1,2,3,5,4 ระบบย่อยที่ 1,2,3 จะเป็นระบบย่อยที่อยู่ในกลุ่มที่ 1 ส่วนระบบย่อย 5 และ 4 อยู่ในกลุ่มที่ 2 ในกรณีที่ ระบบย่อยที่ใช้ข้อมูล อยู่ทางด้านขวาของระบบย่อยที่ให้กำเนิดข้อมูล ก็จะมีวิธีการเรียงระบบย่อยในลักษณะเดียวกัน เช่น รูปที่ 4.17 ก็จะสามารถลำดับระบบย่อยที่ใช้ข้อมูลได้ดังนี้ 6,8,9,7



รูปที่ 4.17 รูปแสดงตัวอย่างตำแหน่งของระบบย่อย (2)

2. ข้อมูลที่ใช้เมื่อมีเส้นความสัมพันธ์ลากเข้าหาระบบย่อย

2.1 ข้อมูลจำนวนระบบย่อยที่ให้กำเนิดข้อมูลที่อยู่ทางด้านบน หรืออยู่ทางด้านล่างของระบบย่อยที่ใช้ข้อมูลอย่างใดอย่างหนึ่ง โดยดูจากตำแหน่งบนจอภาพ เพื่อใช้คำนวณตำแหน่งจุด ที่เส้นความสัมพันธ์ถูกลากเข้าหาระบบย่อยที่ใช้ข้อมูล เช่น ถ้าทางด้านบนของระบบย่อย ที่นำข้อมูลไปใช้ มีระบบย่อยที่ให้กำเนิดข้อมูลอยู่ 4 ระบบ และ ถ้าด้านในแนวระดับของสี่เหลี่ยมที่แทนระบบมีขนาด 400 จุด วิธีการคำนวณระยะห่างระหว่างจุดคือ

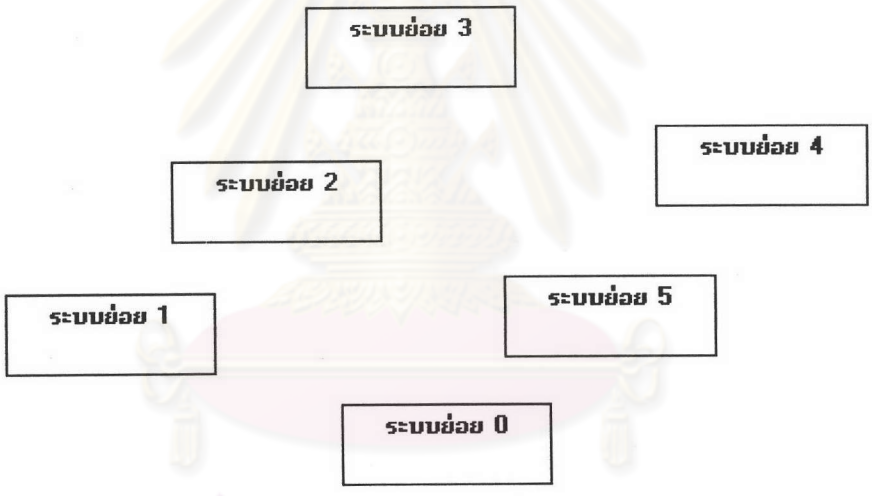
ความยาวของด้านในแนวระดับ / (จำนวนระบบย่อยที่ให้กำเนิดข้อมูล + 1)

ดังนั้นแต่ละจุดที่เป็นจุดสิ้นสุดของเส้นความสัมพันธ์ จะมีระยะห่าง เท่ากับ

400/(4+1) = 80 จุด หมายความว่า จุดแรกที่ลากเส้นความสัมพันธ์เข้าหาจะอยู่ห่างจากมุมซ้ายของสี่เหลี่ยมไปทางขวาเป็นระยะ 80 จุด และเส้นความสัมพันธ์เส้นที่สอง จะอยู่จากมุมของสี่เหลี่ยมไปทางขวาเป็นระยะ 160 จุด ถ้าระบบย่อยที่ให้กำเนิดข้อมูลอยู่ทางด้านล่าง ก็มีวิธีการคำนวณอย่างเดียวกัน

2.2 ข้อมูลลำดับของระบบย่อย ที่จะลากเส้นความสัมพันธ์เข้าหา เป็นการเลือกระบบย่อย ที่จะต้องนำจุดต่าง ๆ ในแนวระดับของสี่เหลี่ยมที่คำนวณได้ในหัวข้อ 2.1 ซึ่งเป็นจุดที่เส้นความสัมพันธ์ ถูกลากเข้าหาระบบที่นำข้อมูลไปใช้ โดยการเรียงระบบย่อยที่ใช้ข้อมูล มีวิธีการเรียงดังนี้

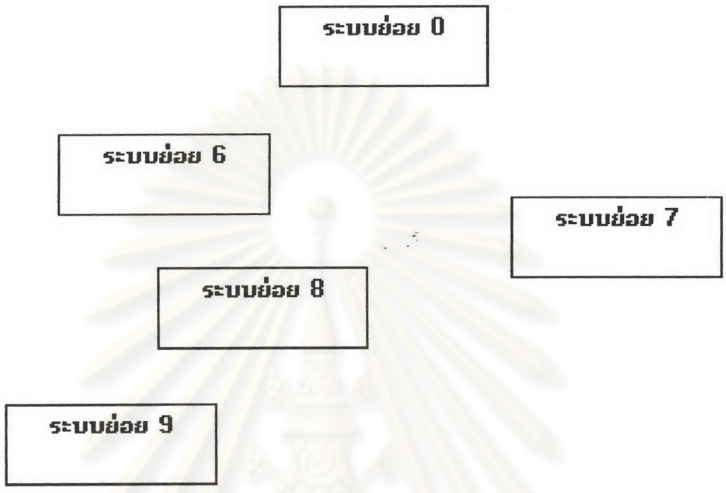
2.2.1 แบ่งระบบย่อยที่ให้กำเนิดข้อมูลเป็น 2 กลุ่ม โดยแบ่งตามตำแหน่งบนจอภาพ ถ้าระบบย่อยที่ให้กำเนิดข้อมูล อยู่ทางด้านซ้ายของระบบย่อยที่ใช้ข้อมูล จัดให้อยู่ในกลุ่มที่ 1 ถ้าอยู่ทางด้านขวาของระบบย่อยที่ให้กำเนิดข้อมูล ให้จัดอยู่ในกลุ่มที่ 2 ถ้ามีความสัมพันธ์ตามรูปที่ 4.18 (ระบบย่อย 0 คือระบบที่นำข้อมูลไปใช้) ระบบย่อยที่ 1,2,3 จะอยู่กลุ่มที่ 1 ส่วนระบบย่อยที่ 4,5 จะอยู่กลุ่มที่ 2



รูปที่ 4.18 รูปแสดงตัวอย่างตำแหน่งของระบบย่อย (3)

2.2.2 ลำดับระบบย่อยที่ให้กำเนิดข้อมูล เพื่อเรียงการลากเส้นความสัมพันธ์จากซ้ายไปขวา โดยจะเรียงในกลุ่มที่ 1 ก่อน (กลุ่มระบบย่อยที่ให้กำเนิดข้อมูล อยู่ทางซ้ายของระบบย่อยที่ใช้ข้อมูล) โดยหลักการเรียงก็คือ ระบบย่อยที่ให้กำเนิดข้อมูล อยู่ใกล้กับระบบย่อยที่ใช้ข้อมูลมากที่สุด โดยคำนึงถึงความใกล้ในแนวดิ่งเป็นเกณฑ์ จัดให้เป็นลำดับที่ 1 ส่วนระบบย่อยอื่นที่ ห่างออกไปก็จัดเป็นลำดับที่ 2,3,4 ... N ไปเรื่อย ๆ (N แทนจำนวนระบบย่อยที่ให้กำเนิดข้อมูล ในกลุ่มที่ 1) ในกลุ่มที่ 2 ก็เช่นกัน (กลุ่มระบบย่อยที่ให้กำเนิดข้อมูล อยู่ทางด้านขวาของระบบย่อยที่ใช้ข้อมูล) คือ ระบบย่อยที่ให้กำเนิดข้อมูล อยู่ใกล้กับระบบย่อยที่ใช้ข้อมูลมากที่สุด โดยคำนึงถึงความใกล้ในแนวดิ่งเป็นเกณฑ์ จัดให้เป็นลำดับที่ N+1 ส่วนระบบย่อยอื่นที่ห่างออกไป ก็จัดเป็นลำดับที่ N+2,N+3,N+4 ... ดู

ตัวอย่างจากรูป 4.18 จะลำดับระบบย่อยได้ดังนี้ คือ 1,2,3,5,4 ระบบย่อยที่ 1,2,3 จะเป็นระบบย่อยที่อยู่ในกลุ่มที่ 1 ส่วนระบบย่อย 5 และ 4 อยู่ในกลุ่มที่ 2 ในกรณีที่ ระบบย่อยที่ให้กำเนิดข้อมูลอยู่ทางด้านล่างของระบบย่อยที่ใช้ข้อมูล ก็จะมีวิธีการเรียงระบบย่อยในลักษณะเดียวกัน เช่น รูปที่ 4.19 ก็จะสามารถ ลำดับระบบย่อยที่ใช้ข้อมูลได้ดังนี้ คือ 6,8,9,7



รูปที่ 4.19 รูปแสดงตัวอย่างตำแหน่งของระบบย่อย (4)

<u>การเขียนโปรแกรมด้วยวิชวลเบสิก</u>

โครงสร้างข้อมูลที่ใช้เก็บความสัมพันธ์ระหว่างระบบย่อยที่ใช้ข้อมูล กับระบบย่อยที่ให้กำเนิดข้อมูล เป็นตัวแปรโครงสร้าง มีการประกาศตัวแปรโครงสร้างดังนี้

```
Type relunit
    unit As Integer
    pointto As Integer
End Type
```

ถ้ามีระบบย่อยทั้งหมด M ระบบ จะประกาศใช้ตัวแปรอะเรย์เท่ากับ M*(2+1+1+6) ประกาศเป็นตัวแปรชุดดังนี้

```
Dim a ( M*10 ) As relunit
```

ตำแหน่งต่าง ๆ บนตัวแปรชุดอธิบายด้วยรูป 4.21 มีความหมายดังนี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ช่วงที่ 1 ตั้งแต่ ตัวแปรชุดตำแหน่งที่ 1 ถึง M * 2 จะเก็บจำนวนระบบย่อยที่ใช้ข้อมูลโดยแบ่งไปทางซ้ายและทางขวา และจำนวนระบบย่อยที่ให้กำเนิดข้อมูลโดยแบ่งเป็นบนและล่าง ของแต่ละระบบย่อย

ช่วงที่ 2 ตั้งแต่ ตัวแปรชุดตำแหน่งที่ M * 2 + 1 ถึง M * 2 + M โดย a().unit จะเก็บระบบย่อยที่ใช้ข้อมูล ที่ เป็นลำดับที่ 1 และ a().pointto จะบอกตำแหน่งในตัวแปรชุด ที่เป็นที่เก็บระบบย่อยที่ใช้ข้อมูลเป็นลำดับที่ 2

ช่วงที่ 3 ตั้งแต่ ตัวแปรชุดตำแหน่งที่ M * 3 + 1 ถึง M * 3 + M โดย a().unit จะเก็บระบบย่อยที่ให้กำเนิดข้อมูล ที่เป็นลำดับที่ 1 และ a().pointto จะบอกตำแหน่งในตัวแปรชุด ที่เป็นที่เก็บระบบย่อยที่ให้กำเนิดข้อมูลเป็นลำดับที่ 2

ช่วงที่ 4 ตั้งแต่ ตัวแปรชุด M * 4 + 1 ถึง M * 4 + 6 * M โดย a().unit จะเก็บระบบย่อยที่เป็นลำดับที่ 2 แต่จะเป็นระบบย่อยที่ให้กำเนิดข้อมูล หรือ เป็นระบบย่อยที่ใช้ข้อมูล ขึ้นอยู่กับว่าถูกตัวแปรในช่วง M * 2 + 1 ถึง M * 2 + M หรือ M * 3 + 1 ถึง M * 3 + M และ a().pointto จะบอกตำแหน่งในตัวแปรชุด ที่เป็นที่เก็บระบบย่อยที่เป็นลำดับที่ 3 เป็นต้นไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตำแหน่งใน ตัวแปรชุด ที่ i	a(i).unit	a(i).pointto	
1	**ซ้าย1**	**ขวา1**	ช่วงที่ 1
2	**บน1**	**ล่าง1**	
	.....		
M*3+1	**ux1**	**px1**	ช่วงที่ 2
	.....		
M*3+2	**uy1**	**py1**	ช่วงที่ 3
	.....		
M*4+1	**u1**	**px1**	ช่วงที่ 4
	.....		
M*4+6M	.....		

รูปที่ 4.20 รูปแสดงโครงสร้างข้อมูลของความสัมพันธ์ระหว่างระบบ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย