บทที่ 2

แนวคิดและทฤษฎี

2.1 ทฤษฎีบัญชี

2.1.1 หลักการบัญชีเบื้องต้น

ข้อมูลขั้นพื้นฐานในระบบธุรกิจคือรายการขั้นต้นหรือรายการค้า ที่เกิดจากการบันทึกรายการเกี่ยวกับเหตุการณ์ทางการค้าหรือการดำเนินงานธุรกิจระหว่างหน่วยงานภายในกิจการเองหรือระหว่างคู่ค้า ได้แก่ รายการขาย รายการซื้อ รายการรับชำระเงิน รายการชำระเงิน และรายการอื่นๆ ดังนั้นจึงต้องคำนึงถึงหลักการทางบัญชีเพื่อการบันทึกรายการค้าด้วยความถูกต้องครบถ้วน ดังนี้

2.1.1.1 สมการบัญชี [1]

กิจการหนึ่ง ๆ มีทรัพย์สินที่กิจการเป็นเจ้าของ มีหนี้สินที่เป็นภาระของกิจการซึ่งต้องชำระคืน และมีเงินลงทุนที่ผู้ถือครองอยู่ เมื่อธุรกิจมีกำไรที่ยังไม่ถูกแบ่งปัน ก็จะสะสมไว้เป็นกำไรสะสมซึ่งถือว่าเป็นส่วนหนึ่งของผู้ถือหุ้นด้วย ผลต่างระหว่างสินทรัพย์และหนี้สินขององค์การเรียกว่า ส่วนของผู้ถือหุ้นสามารถเขียนเป็นสมการบัญชีได้ดังนี้

สินทรัพย์ - หนี้สิน = ส่วนของผู้ถือหุ้น

กลุ่มบัญชีส่วนของผู้ถือหุ้น พบว่ามีบัญชีอยู่สองประเภทใหญ่ ๆ ได้แก่ประเภทที่เกี่ยวกับทุนและประเภทที่เกี่ยวกับกำไรหรือขาดทุนจากการดำเนินธุรกิจ กำไรหรือขาดทุนเป็นผลต่างระหว่างรายได้และต้นทุนรวมรายจ่าย ผลกำไรทำให้ส่วนของผู้ถือหุ้นเพิ่มขึ้น ขาดทุนทำให้ส่วนของผู้ถือหุ้นลดลงเนื่องจากรายการที่เกี่ยวกับรายได้และรายจ่ายรวมต้นทุน ถือเป็นรายการค้าที่ต้องลงบัญชีด้วย จึงกำหนดกลุ่มบัญชีขึ้นอีก 2 กลุ่ม ได้แก่ บัญชีรายได้ และบัญชีต้นทุนรวมรายจ่าย แล้วเขียนสมการบัญชีใหม่ได้ดังนี้

สินทรัพย์ = หนี้สิน + ส่วนที่เกี่ยวกับทุน + (รายได้ - ต้นทุนและค่าใช้จ่าย)

เขียนสมการใหม่ได้ดังนี้

สินทรัพย์ + ต้นทุนและรายจ่าย = หนี้สิน + ส่วนที่เกี่ยวกับทุน + รายได้

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.1.1.2 การบันทึกรายการตามหลักการบัญชีคู่ (Double - Entry Book Keeping) และการบันทึกรายการทางด้านเดบิตและเครดิต [2]

หลักการบัญชีที่เป็นที่ยอมรับและใช้บันทึกรายการกันแพร่หลายที่สุดในทุกประเทศ คือ หลักการบัญชีคู่ ความหมายของหลักการบัญชีคู่ คือ เมื่อมีรายการทางการค้าเกิดขึ้นทุกรายการต้องนำมาบันทึกไว้ในบัญชีสองด้านคือ ทางด้านเดบิต และทางด้านเครดิต ด้วยจำนวนเงินที่เท่ากันทั้งสองด้าน แต่จำนวนบัญชีที่ลงนั้นไม่จำเป็นต้องเท่ากัน กล่าวคือ อาจบันทึกทางด้านเดบิตเพียงบัญชีเดียว แต่บันทึกทางด้านเครดิตสองหรือสามบัญชีก็ได้

2.1.1.3 ระบบการบันทึกรายการทางด้านเดบิตและเครดิต (The system of debits and credits) ลักษณะโดยทั่วไปของการบันทึกรายการทางการบัญชีคือ จะมีการบันทึกรายการที่เพิ่มขึ้น และรายการที่ลดลง

การบันทึกรายการทางบัญชีมีเรื่องที่ต้องทำความเข้าใจอยู่ 5 หัวข้อ ด้วยกันคือ

2.1.1.3.1 รายการทางด้านเดบิต (Debit) จะหมายถึงทางด้านซ้ายมือของบัญชีทุกประเภทและรายการทางด้านเครดิต (Credit) จะอยู่ทางด้านขวาของทุกบัญชี

2.1.1.3.2 บัญชีสินทรัพย์และบัญชีค่าใช้จ่ายจะเพิ่มขึ้นทางด้านเดบิตเสมอ ส่วนบัญชีที่เกี่ยวข้องกับหนี้สินทุนและบัญชีรายได้จะเพิ่มขึ้นทางด้านเครดิตเท่านั้น

2.1.1.3.3 บัญชีสินทรัพย์และบัญชีค่าใช้จ่ายจะลดลงทางด้านเครดิตและบัญชีหนี้สิน บัญชีทุน ตลอดจนบัญชีรายได้จะลดลงทางด้านเดบิต

2.1.1.3.4 ยอดคงเหลือตามปกติของบัญชีแยกประเภทเกี่ยวกับสินทรัพย์จะคงเหลือทางด้านเดบิต ส่วนบัญชีแยกประเภทเกี่ยวกับหนี้สินและทุนจะมียอดคงเหลือทางด้านเครดิต

2.1.1.3.5 บัญชีแยกประเภทรายได้ จะมียอดรวมทางด้านเครดิตและบัญชีแยกประเภทเกี่ยวกับค่าใช้จ่ายจะมียอดรวมทางด้านเดบิต

	สินทรัพย์ Assets		หนี้สิน Liabilities		ส่วนของเจ้าของ Owners Equity		รายได้ Revenue		รายจ่าย Expenses	
	เดบิต	เครดิต	เดบิต	เครดิต	เดบิต	เครดิต	เดบิต	เครดิต	เดบิต	เครดิต
รายการเพิ่มขึ้น	+			+		+		+	+	
รายการลดลง		-	-		-		-			-
ยอดคงเหลือ	*			*		*				

รูปที่ 2.1 แสดงรูปแบบรายการเพิ่มขึ้นลดลง รายการเดบิต เครดิต และยอดคงเหลือของบัญชีแยกประเภททั่วไป

2.1.1.4 รหัสบัญชี

รหัสบัญชีหรือเลขที่บัญชี [3] คือตัวเลขหรือตัวอักษรที่ใช้แทนชื่อบัญชี

การจัดหมวดหมู่และการให้รหัสบัญชีโดยทั่วไปที่นิยมกันคือใช้ตัวเลขหลักแรกแสดงประเภทของบัญชีเป็นหมวดหมู่ใหญ่ที่สุด [1]

2.1.1.4.1 บัญชีแยกประเภทสินทรัพย์เริ่มต้นด้วยเลข 1

2.1.1.4.2 บัญชีแยกประเภทหนี้สินเริ่มต้นด้วยเลข 2

2.1.1.4.3 บัญชีแยกประเภทส่วนของเจ้าของเริ่มต้นด้วยเลข 3

2.1.1.4.4 บัญชีแยกประเภทรายได้เริ่มต้นด้วยเลข 4

2.1.1.4.5 บัญชีแยกประเภทรายจ่ายเริ่มต้นด้วยเลข 5

และหลักถัดไปแสดงถึงหมวดหมู่รองลงไปตามลำดับ

2.1.2 การบัญชีกิจการผลิตสินค้า

กิจการที่ทำการผลิตสินค้าเองจะต้องมีการซื้อวัตถุดิบเพื่อแปรสภาพให้เป็นสินค้าสำเร็จรูป ต้องจ่ายค่าแรงคนงานและจ่ายค่าใช้จ่ายในการผลิต ทั้งสามสิ่งนี้เรียกว่า ปัจจัยการผลิต กิจการที่ทำการผลิตสินค้าต้องมีการบันทึกรายการเกี่ยวกับต้นทุนการผลิต โดยเริ่มตั้งแต่การซื้อวัตถุดิบ การจัดสรรวัตถุเข้างาน การจ่ายค่าแรงงานและการจ่ายค่าใช้จ่ายในการผลิต เมื่อบันทึกต้นทุนปัจจัยการผลิตแล้ว ขั้นตอนต่อไปคือ การจัดสรรปัจจัยการผลิตเพื่อเข้าสู่งานการผลิต ผลิตเสร็จจะบันทึกการขาย ต้นทุนขายและค่าใช้จ่ายในการดำเนินงาน จนครบวงจรทางการบัญชี [4]

2.1.2.1 ระบบบัญชีสำหรับกิจการที่ทำการผลิต [5] นั้นแบ่งออกได้เป็น 2 ส่วนคือ

2.1.2.1.1 บัญชีการเงิน (Financial Accounting) ซึ่งประกอบด้วยระบบบัญชีสำหรับการขาย การซื้อ เงินสดรับ เงินสดจ่าย และเงินเดือน

2.1.2.1.2 บัญชีต้นทุน (Cost Accounting) คือระบบบัญชีซึ่งใช้ในการรวบรวมข้อมูลเกี่ยวกับต้นทุนของสินค้าที่ทำการผลิต อันได้แก่ วัตถุดิบ ค่าแรง และค่าใช้จ่ายโรงงาน ซึ่งในระบบบัญชีต้นทุนนั้นจะต้องทำขึ้นเพื่อที่จะให้ข้อมูลในการคำนวณหาต้นทุนต่าง ๆ ดังต่อไปนี้

1) ต้นทุนต่อหน่วยของวัตถุดิบที่ใช้ในการผลิต

2) ต้นทุนต่อหน่วยของค่าแรง

3) ต้นทุนต่อหน่วยของค่าใช้จ่ายโรงงาน

2.1.2.2 วงจรการบันทึกต้นทุนการผลิต [6]

การบันทึกรายการบัญชีของกิจการที่ทำการผลิตสินค้าจะมีขั้นตอนและรายละเอียดมากกว่าของกิจการที่ซื้อสินค้าที่ผลิตสำเร็จรูปแล้วมาขายต่อ เพราะในการผลิตสินค้าต้องมีปัจจัยการผลิตประกอบด้วย วัตถุดิบ ค่าแรง และค่าใช้จ่ายในการผลิตมาไว้เพื่อนำเข้าสู่กระบวนการผลิต มีการสำรวจสินค้าที่ยังผลิตไม่เสร็จ ว่าประกอบด้วยวัตถุดิบ ค่าแรง และค่าใช้จ่ายในการผลิตเป็นจำนวนรวมกันทั้งสิ้นเท่าไร เพื่อนำมาขึ้นบัญชีงานระหว่างทำ เป็นต้น

กระบวนการบันทึกบัญชี มีขั้นตอนดังนี้

2.1.2.2.1 การบันทึกต้นทุนการจัดหาปัจจัยการผลิต

ปัจจัยการผลิตประกอบด้วย วัตถุดิบ ค่าแรงและค่าใช้จ่ายในการผลิต เมื่อจัดหาปัจจัยทั้ง 3 อย่างได้แล้ว การบันทึกรายการทำได้ดังนี้

1) เมื่อซื้อวัตถุดิบ

เดบิต บัญชีวัตถุดิบ	xxx	
เครดิต เงินสด/เจ้าหนี้-การค้า		xxx

2) เมื่อมีการส่งคืนสินค้า

2.1) มีการส่งคืนวัตถุดิบ จะทำให้ยอดคงเหลือลดลง

เดบิต บัญชีเงินสด/เจ้าหนี้	xxx	
เครดิต บัญชีวัตถุดิบ		xxx

2.2) มีการให้ส่วนลดแทนการส่งคืนสินค้า จะทำให้ต้นทุนต่อหน่วยลดลง

เดบิต บัญชีเงินสด/เจ้าหนี้	xxx	
เครดิต บัญชีวัตถุดิบ		xxx

3) ค่าขนส่งเมื่อซื้อ

3.1) ในกรณีที่ผู้ซื้อและผู้ขายตกลงซื้อขายตามราคา FOB destination คือผู้ขายรับขนวัตถุดิบให้ถึงปลายทางที่ตกลงกัน กรณีเช่นนี้ผู้ขายจะรวมค่าขนส่งไว้ในใบกำกับสินค้าเป็นต้นทุนวัตถุดิบแล้ว

เดบิต บัญชีวัตถุดิบ	xxx	
เครดิต บัญชีเงินสด/เจ้าหนี้		xxx

3.2) ในกรณีที่ผู้ซื้อและผู้ขายตกลงซื้อขายตามราคา FOB shipping point ที่ผู้ขายขนวัตถุดิบไปถึงสถานีหรือ บุคคลที่รับขนส่งสินค้าเท่านั้น ผู้ซื้อต้องจ่ายค่าขนส่งเมื่อนำวัตถุดิบจากสถานที่ขนส่งมายังโรงงานที่ผลิตสินค้า

เดบิต บัญชีค่าขนส่งเมื่อซื้อ	xxx	
เครดิต บัญชีเงินสด/เจ้าหนี้		xxx

4) เมื่อมีการจ่ายค่าแรง

เดบิต บัญชีค่าแรง	xxx	
เครดิต บัญชีเงินสด		xxx

5) ค่าใช้จ่ายในการผลิต

ค่าใช้จ่ายในการผลิตประกอบด้วย วัตถุดิบทางอ้อม ค่าแรงทางอ้อม ค่าซ่อมแซมอาคารและอุปกรณ์ ค่าเบี้ยประกัน เครื่องจักร และอุปกรณ์ ค่าภาษี ค่าสาธารณูปโภค ค่าเสื่อมราคาสินทรัพย์ ฯลฯ จะใช้การบันทึกรายการโดยใช้บัญชีคุมยอดและบัญชีแยกประเภทย่อยโดยใช้ชื่อบัญชี

คุมยอดค่าใช้จ่ายในการผลิตและบัญชีย่อย เช่น บัญชีย่อยวัตถุดิบทางอ้อม ค่าแรงทางอ้อม ค่าซ่อมแซมอาคารและอุปกรณ์ ฯลฯ เป็นต้น

2.1.2.2.2 การบันทึกต้นทุนปัจจัยการผลิตเข้างาน

เมื่อกิจการจัดหาปัจจัยการผลิตได้ตามความต้องการแล้วกิจการก็จะเริ่มทำการผลิตโดยนำปัจจัยการผลิตเข้าสู่กระบวนการผลิต การบันทึกต้นทุนตามระบบต่อเนื่องกิจการจะมีบัญชีวัตถุดิบทางตรง (คงเหลือต้นงวด) บัญชีงานระหว่างทำ (คงเหลือต้นงวด) เมื่อมีการซื้อวัตถุดิบแทนที่จะบันทึกในบัญชีซื้อวัตถุดิบจะนำไปบันทึกในบัญชีวัตถุดิบทางตรง เมื่อมีการจ่ายค่าแรงก็จะบันทึกในบัญชีค่าแรงทางตรง เมื่อมีการจ่ายค่าใช้จ่ายในการผลิตแทนที่จะเปิดบัญชีย่อยค่าใช้จ่ายทุกประเภทกิจการจะมีบัญชีคุมยอดและบัญชีย่อยค่าใช้จ่ายแต่ละประเภทคือจะมีบัญชีคุมยอดค่าใช้จ่ายการผลิตและจะมีบัญชีแยกประเภทย่อยค่าใช้จ่ายการผลิต เช่น ค่าวัตถุดิบทางอ้อม ค่าแรงทางอ้อม ค่าซ่อมแซมโรงงาน ค่าเสื่อมราค่าสินทรัพย์แต่ละชนิด เป็นต้น

เมื่อนำปัจจัยการผลิตสู่กระบวนการผลิต จะนำปัจจัยการผลิตเข้าบัญชีงานระหว่างทำทั้งหมด โดยมีขั้นตอนดังนี้

1) การเบิกวัตถุดิบทางตรงเข้าสู่การผลิต เมื่อได้รับอนุมัติให้เบิกวัตถุดิบทางตรงเข้าสู่การผลิตได้แล้ว การบันทุกรายการทำได้ดังนี้

เดบิต บัญชีงานระหว่างทำ	xxx	
เครดิต บัญชีวัตถุดิบทางตรง		xxx

2) คำนวณค่าแรงเข้าสู่การผลิต

เดบิต บัญชีงานระหว่างทำ	xxx	
เครดิต บัญชีค่าแรงทางตรง		xxx

3) โอนค่าใช้จ่ายการผลิตเข้าสู่การผลิต

เดบิต บัญชีงานระหว่างทำ	xxx	
เครดิต บัญชีคุมยอดค่าใช้จ่ายการผลิต		xxx

2.1.2.2.3 บันทึกต้นทุนสินค้าที่ผลิตเสร็จเพื่อนำเข้าเก็บในคลังสินค้า

เมื่อนำปัจจัยการผลิตเข้าสู่บัญชีงานระหว่างทำแล้ว เมื่อทำการผลิตเสร็จเรียบร้อยแล้ว จะมีต้นทุนสินค้าที่ผลิตเสร็จจะทำการโอนต้นทุนนี้เข้าสู่บัญชีสินค้าสำเร็จรูป (Finished Goods Inventory Account) การบันทึกรายการทำได้ดังนี้

เดบิต บัญชีสินค้าสำเร็จรูป/ต้นทุนสินค้าสำเร็จรูป	xxx	
เครดิต บัญชีงานระหว่างทำ		xxx

ถ้ากิจการผลิตสินค้าหลายประเภท บัญชีสินค้าสำเร็จรูปนี้จะเป็นบัญชีคุมยอดสินค้าและมีบัญชีแยกประเภทย่อยคือบัตรสินค้า

2.1.2.2.4. บันทึกต้นทุนสินค้าที่ขายได้

กิจการที่ทำการผลิตสินค้าส่วนใหญ่จะบันทึกรายการสินค้าแบบต่อเนื่อง ดังนั้นเมื่อผลิตเสร็จจะมีบัญชีสินค้าสำเร็จรูปคุมยอดของสินค้าที่ผลิตเสร็จแล้ว เมื่อนำออกขายจะกระบวนการบันทึกรายการที่เกี่ยวข้องดังนี้

1) เมื่อมีการขาย

คู่ที่ 1 บันทึกต้นทุนขายและสินค้าคงเหลือในราคาทุน

เดบิต บัญชีต้นทุนสินค้าขาย (ราคาทุน)

เครดิต บัญชีสินค้าสำเร็จรูป (ราคาทุน)

คู่ที่ 2 บันทึกบัญชีขายและบัญชีลูกหนี้การค้า/เงินสด ในราคาขาย

เดบิต บัญชีลูกหนี้-การค้า/เงินสด (ราคาขาย)

เครดิต บัญชีขาย (ราคาขาย)

2) เมื่อมีการรับคืนสินค้า

2.1) เมื่อมีการรับคืนตัวสินค้า จะทำให้ยอดสินค้าคงเหลือเพิ่มขึ้น ต้นทุนสินค้าที่ขายจะลดลง

คู่ที่ 1 บันทึกตามราคาทุน

เดบิต บัญชีสินค้าสำเร็จรูป (ราคาทุน)

เครดิต บัญชีต้นทุนสินค้าขาย (ราคาทุน)

คู่ที่ 2 บันทึกตามราคาขาย

เดบิต บัญชีสินค้ารับคืนและส่วนลด (ราคาขาย)

เครดิต บัญชีลูกหนี้การค้า/เงินสด (ราคาขาย)

2.2) เมื่อมีการให้ส่วนลดแทนการรับคืนตัวสินค้า

เดบิต บัญชีสินค้ารับคืนและส่วนลด xxx

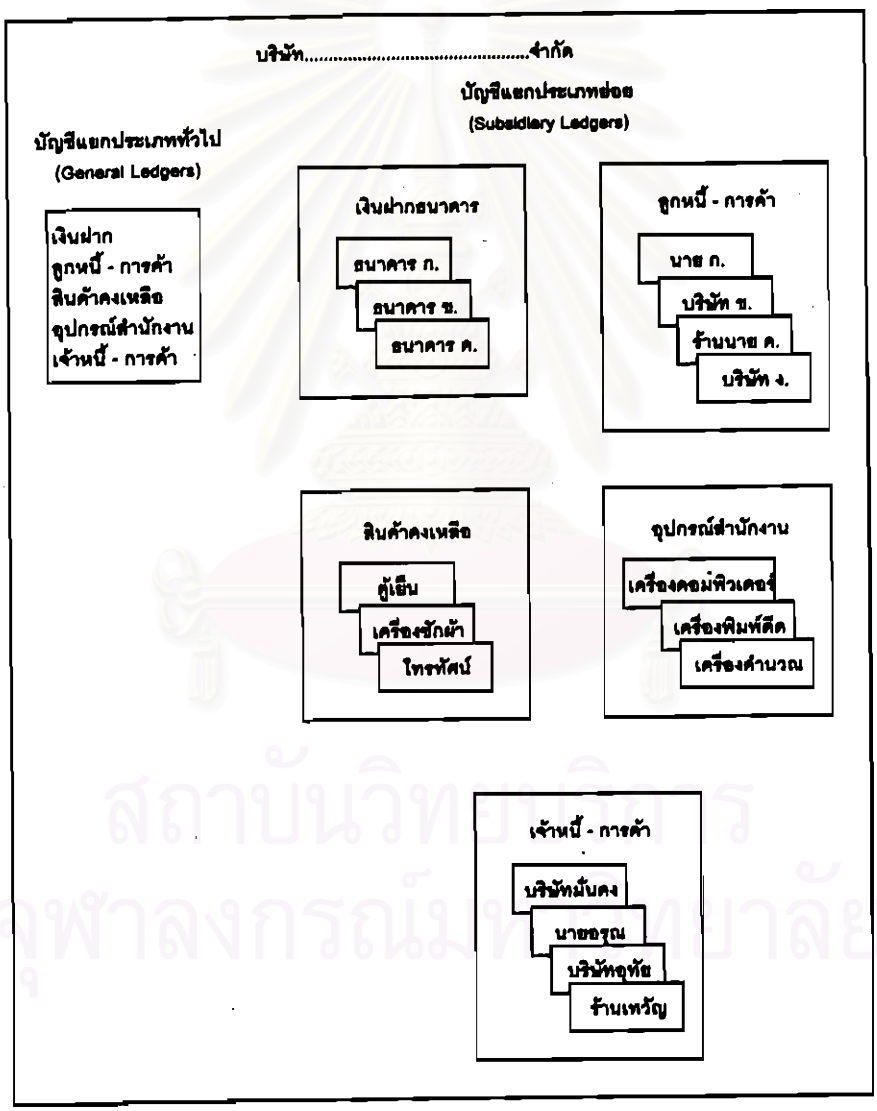
เครดิต บัญชีลูกหนี้-การค้า/เงินสด xxx

2.1.3 **ความสัมพันธ์ระหว่างบัญชีคุมยอด** (Control Account) **และบัญชีย่อยหรือบัญชีแยกประเภทย่อย** (Subsidiary Ledger)

ในกิจการขนาดใหญ่อาจจะมีรายการประเภทหนึ่งประเภทใดเกิดขึ้นจำนวนมากอันจะทำให้จำนวนบัญชีที่เปิดขึ้นในบัญชีแยกประเภททั่วไปมีมาก [5] ซึ่งทำให้ไม่สะดวกในการปิดบัญชีเพื่อทำงบการเงิน ดังนั้นจึงมีการแก้ไขโดยใช้บัญชีคุมยอดซึ่งจะนำแต่เพียงยอดรวมของรายการประเภทนั้นเมื่อสิ้นเดือนไปลงบัญชีคุมยอดส่วนรายละเอียดต่าง ๆ จะนำไปลงในบัญชีย่อย ซึ่งเป็นบัญชีประกอบบัญชีคุมยอด ตัวอย่างเช่น ในกรณีที่กิจการมีลูกหนี้หลายร้อยรายแทนที่จะเปิดบัญชีลูกหนี้รายตัวหลายร้อยบัญชีในบัญชีแยกประเภททั่วไป กิจการก็จะบันทึกรายการเกี่ยวกับลูกหนี้รายตัวเหล่านี้ในบัญชีย่อยและเก็บยอดรวมไปลงบัญชีคุมยอดลูกหนี้ (Accounts Receivable Control) เพียงบัญชีเดียว ทางด้านเดบิตของบัญชีคุมยอดลูกหนี้จะบันทึกแต่เพียงยอดรวมของลูกหนี้ทั้งหมดยอดเดียว ส่วนที่เป็นลูกหนี้รายตัวจะปรากฏอยู่ในบัญชีย่อยในทำนองเดียวกันทางด้าน

เครดิตของบัญชีคุมยอดลูกหนี้ก็จะบันทึกแต่เพียงยอดรวมของรายการที่นำไปลงทางด้านเครดิตของบัญชีลูกหนี้รายตัวในบัญชีย่อย

สรุปได้ว่าการจัดทำบัญชีย่อยก็คือการเก็บรายละเอียดของข้อมูลทางการค้าที่เกิดขึ้นจากระบบงาน ซึ่งเป็นข้อมูลที่เกิดขึ้นประจำและมีจำนวนมาก เพื่อผลประโยชน์ทางการค้าที่จะได้รับเช่น การใช้ข้อมูลจากระบบงานลูกหนี้ในการเร่งรัดหนี้สิน หรือจากระบบงานเจ้าหนี้เช่น การจ่ายหนี้ได้ภายในระยะเวลาที่เหมาะสม ลักษณะของบัญชีย่อยอาจอยู่ในรูปบัญชีรูปตัวทีหรือบัตร (การ์ด) ก็ได้ [4] การจัดทำบัญชีคุมยอดก็เพื่อทำการสรุปผลและใช้เป็นบัญชีแยกประเภททั่วไปเพื่อลดจำนวนของข้อมูลที่จะเกิดในรายงานการเงินต่าง ๆ ของระบบงานบัญชีแยกประเภท



รูปที่ 2.2 แสดงความสัมพันธ์ระหว่างบัญชีแยกประเภททั่วไป และบัญชีแยกประเภทย่อย [2]

2.1.4 บัญชีคุมยอดและบัญชีย่อย (Control and Subsidiary Accounts) ของธุรกิจผลิตและจำหน่าย [7]

จากวงจรการผลิตและระบบงานย่อยที่จัดทำในธุรกิจผลิตและจำหน่ายทำให้เกิดบัญชีคุมยอดและบัญชีย่อยที่สำคัญ ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 แสดงบัญชีคุมยอดและบัญชีย่อยในธุรกิจผลิตและจำหน่าย

บัญชีคุมยอด	บัญชีย่อย
1. บัญชีคุมยอดลูกหนี้	บัญชีย่อยตามลูกหนี้รายบุคคล
2. บัญชีคุมยอดเจ้าหนี้	บัญชีย่อยตามเจ้าหนี้รายบุคคล
3. บัญชีคุมยอดวัตถุดิบ	บัตรวัตถุดิบ - ใช้ระบบการตรวจนับแบบต่อเนื่อง (Perpectual inventory Sysem)
4. บัญชีคุมยอดค่าแรง	บันทึกเงินได้ประจำตัวคนงานแต่ละคนหรือใบสรุปค่าแรงประจำแผนก
5. บัญชีคุมยอดค่าใช้จ่ายการผลิต	ใบสรุปค่าใช้จ่ายประจำแผนก
6. บัญชีคุมยอดงานระหว่างทำ	งบต้นทุนการผลิตประจำแผนก
7. บัญชีคุมยอดสินค้าสำเร็จรูป	บัตรสินค้าสำเร็จรูป
8. บัญชีคุมยอดเครื่องจักรและอุปกรณ์	บัญชีย่อยตามประเภทเครื่องจักรและอุปกรณ์หรือบัตรประจำเครื่องจักรและอุปกรณ์

2.2 การพัฒนาระบบด้วยเทคโนโลยีเชิงวัตถุ

2.2.1 เทคโนโลยีเชิงวัตถุ [8]

เทคโนโลยีเชิงวัตถุเป็นแนวคิดของการพัฒนาโปรแกรมแบบหนึ่ง โดยผนึก (Encapsulation) รวมข้อมูลและวิธีการทำงานเข้าไว้ด้วยกันเป็นวัตถุ (Object) และใช้ส่วนติดต่อ (Interface) ในการสื่อสารระหว่างวัตถุต่าง ๆ เมื่อวัตถุตัวใดต้องการใช้ข้อมูลหรือบริการจากอีกวัตถุหนึ่งจะต้องติดต่อผ่านส่วนติดต่อนี้เท่านั้น ซึ่งพบว่าเป็นลักษณะการซ่อนหรือป้องกันการเข้าถึงข้อมูลของวัตถุ (Information hiding) ซึ่งจะทำให้เมื่อมีการเปลี่ยนแปลงในวัตถุหนึ่งจะส่งผลกระทบต่อวัตถุอื่น ๆ น้อยมาก ดังนั้นจึงเป็นเทคโนโลยีที่เป็นแนวทางในการแก้ปัญหาการพัฒนาโปรแกรมในปัจจุบันคือ ปัญหาด้านการใช้ต้นทุนในการสร้าง และบำรุงรักษาจำนวนมาก การขาดความสามารถในการปรับเปลี่ยนการทำงานของโปรแกรมให้เข้ากับความต้องการใหม่ของธุรกิจ

แม้ว่าจุดเริ่มต้นของเทคโนโลยีเชิงวัตถุมุ่งเน้นไปในการแก้ปัญหาทางด้านการเขียนโปรแกรม แต่ในปัจจุบันเริ่มเป็นที่ตระหนักถึงสิ่งที่สำคัญของการเลือกใช้วิธีการ ความสัมพันธ์อย่างมีแบบแผนเพื่อพัฒนา

งานและการวิเคราะห์และออกแบบเชิงวัตถุที่ดีเพื่อนำไปสู่การพัฒนาระบบงานด้วยการโปรแกรมเชิงวัตถุที่มีประสิทธิภาพมากยิ่งขึ้น

2.2.2 **วงจรการพัฒนาซอฟต์แวร์เชิงวัตถุ** [9]

วงจรการพัฒนาซอฟต์แวร์เชิงวัตถุ เป็นวงจรของการทำงานเพิ่ม (Incremental) และการทำซ้ำ (Iterative) ของกระบวนการ วงจรการพัฒนาซอฟต์แวร์เชิงวัตถุ ประกอบด้วยกระบวนการดังนี้

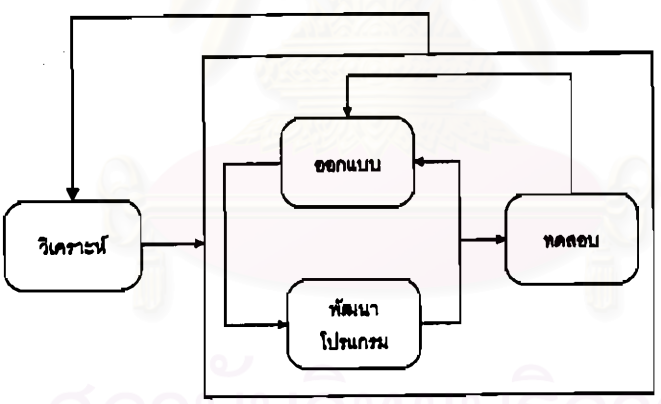
2.2.2.1 การวิเคราะห์เชิงวัตถุ

2.2.2.2 การออกแบบเชิงวัตถุ

2.2.2.3 การพัฒนาโปรแกรมเชิงวัตถุ

2.2.2.4 การทดสอบระบบ

โครงสร้างหลักของวงจรนี้เป็นการวนซ้ำ (Loop) ของการออกแบบและการพัฒนาโปรแกรมเชิงวัตถุ ในส่วนกลางของโครงสร้างซึ่งแสดงถึงพื้นฐานของวงจรในการเปลี่ยนแปลงการพัฒนาโปรแกรมได้อย่างรวดเร็ว เป็นการสร้างโปรแกรมประยุกต์รุ่นใหม่เพื่อทำการประเมินผลและทำการคืนสู่วงจรเพื่อทำการออกแบบซอฟต์แวร์เพิ่มเติมต่อไป ซึ่งการทำซ้ำจะของกระบวนการในวงจรจะนำไปสู่การลดความเสี่ยงในการพัฒนาซอฟต์แวร์



รูปที่ 2.3 แสดงวงจรการพัฒนาซอฟต์แวร์เชิงวัตถุ [9]

2.2.2.1 **การวิเคราะห์งานเชิงวัตถุ**

วัตถุ (Object) คือสิ่งใด ๆ เป็นจริงหรือนามธรรมซึ่งเกี่ยวกับข้อมูลที่จัดเก็บและวิธีการในการจัดการกับข้อมูลนั้น [10]

คลาส (Class) คือการรวบรวมวัตถุหรือกลุ่มของวัตถุด้วยกลุ่มของลักษณะประจำและบริการ รวมทั้งการบรรยายว่าจะสร้างวัตถุจากคลาสนั้นอย่างไร [8]

ขั้นตอนในการวิเคราะห์งานเชิงวัตถุแบ่งเป็น 2 ขั้นตอนคือ

2.2.2.1.1 การวิเคราะห์ความต้องการระบบ (System Requirments Analysis) เป็นการศึกษาและวิเคราะห์ระบบงานเพื่อทำความเข้าใจกับระบบที่ทำการพัฒนา โดยคำนึงถึงการเชื่อมโยงการทำงานภายในระบบงาน โดยใช้แผนภาพลำดับเหตุการณ์ (Usecase Diagram) หรือแบบจำลองลำดับรายการเปลี่ยนแปลง (Transaction Sequence Model) เป็นเครื่องมือในการบรรยายถึงการทำงานสำหรับกระบวนการทำงานทางธุรกิจ (Business process) [11]

2.2.2.1.2 การวิเคราะห์ระบบงาน แบ่งเป็นระดับการทำงานได้ดังนี้

1) ระดับคลาสและวัตถุ (Class and Object layer) ในการวิเคราะห์งานเชิงวัตถุนั้น ต้องเริ่มต้นจากการค้นหาคลาสและวัตถุ ซึ่งวัตถุที่ทำการค้นหานี้เป็นนามธรรม (Abstraction) ในขอบเขตของปัญหาที่กำหนด ดังนั้นจำนวนคลาสในการวิเคราะห์งานขึ้นอยู่กับขอบเขตของปัญหาที่กำหนดขึ้น ผลลัพธ์ที่ได้จากขั้นตอนนี้คือ คลาสซึ่งแสดงโดยแผนภาพคลาส (class diagrams) [9]

กลวิธีในการเลือกวัตถุในส่วนขอบเขตของปัญหาที่กำหนดคือการเลือกจาก [12]

ก. นักแสดง (actor) และผู้มีส่วนร่วม (participants) โดยมองจากบุคคลและองค์การซึ่งมีการกระทำร่วมกันภายในระบบ บทบาทหน้าที่ ภายใต้สิ่งที่กำลังพิจารณาและการวิเคราะห์ว่าแต่ละนักแสดงมีส่วนร่วมกันอย่างไรในแนวทางที่น่าสนใจ

ข. สถานที่(places)โดยมองจากที่ซึ่งมีสิ่งของ (things) พักอยู่หรือสถานที่บรรจุวัตถุ

ค. สิ่งของโดยมองจากวัตถุที่จับต้องได้ ซึ่งใช้อยู่ภายในขอบเขตของปัญหา

2) ระดับการกำหนดโครงสร้าง (Structure layer) เป็นการแสดงขอบเขตของปัญหาและความรับผิดชอบต่อระบบงาน (Problem domain and System's responsibilities) ที่ซับซ้อนให้สามารถเข้าใจได้ง่ายขึ้น โดยผลลัพธ์ที่ได้จากขั้นตอนนี้ประกอบด้วยโครงสร้างดังนี้คือ [9]

2.1) โครงร่างทั่วไป (Generalization - Specialization Structure) เป็นโครงสร้างที่แสดงการสืบทอดคุณสมบัติ (inheritance) ของวัตถุ

2.2) โครงสร้างทุกส่วน (Whole - Part Structure) เป็นโครงสร้างที่แสดงถึงส่วนประกอบที่สัมพันธ์กันของวัตถุแต่ไม่ได้แสดงถึงการสืบทอดคุณสมบัติของวัตถุ ตัวอย่าง เช่น บุคคลมี 1 ร่างกาย มี 2 แขนและประกอบด้วยอวัยวะอื่น ๆ]

3) ระดับการกำหนดซับเจคหรือกลุ่มงาน (Subject layer) เป็นการกระทำที่ช่วยให้ผู้วิเคราะห์งานสามารถมองภาพรวมได้โดยง่าย ซึ่งเหมาะสำหรับการทำงานที่มีโครงสร้างใหญ่และซับซ้อน โดยการรวบรวมกลุ่มของวัตถุเข้าสู่การเป็นนามธรรมระดับสูงขึ้น (higher - level abstraction) [9]

4) ระดับกำหนดลักษณะประจำ (Attribute layer) คือการกำหนดข้อมูลหรือข่าวสารที่สามารถแสดงสถานะของวัตถุ การทำงานในระดับนี้มีการกำหนดลักษณะประจำซึ่งถือได้ว่าเป็นคุณสมบัติเฉพาะของแต่ละคลาส มีการเพิ่มการติดต่อของวัตถุ เพื่อแสดงความสัมพันธ์ระหว่างคลาสและวัตถุ

ผลลัพธ์ที่ได้จากการทำงานระดับนี้คือ แผนภาพแสดงลักษณะประจำ (Attribute Diagram) และแผนภาพที่แสดงถึงการติดต่อระหว่างวัตถุ (Object Connection Diagram) [9]

4.1) สิ่งที่ต้องพิจารณาเพื่อกำหนดลักษณะประจำ [12]

4.1.1) พิจารณาถึงลักษณะทั่ว ๆ ไปที่สามารถบรรยายลักษณะของวัตถุได้

4.1.2) พิจารณาถึงลักษณะเฉพาะที่สามารถระบุลักษณะของวัตถุได้ ภายในขอบเขตของปัญหาที่กำหนด

4.1.3) พิจารณาถึงสิ่งที่ต้องการจะรู้สำหรับลักษณะของวัตถุเพื่อการทำงานเป็นผลสำเร็จของระบบสารสนเทศ (Information system) ซึ่งบางครั้งจะพิจารณาไปถึงผลลัพธ์ที่จะได้เพื่อช่วยพิจารณาถึงข้อมูลเข้า เพราะข้อมูลเข้าที่จำเป็นจะกลายมาเป็นลักษณะประจำของวัตถุได้

4.1.4) พิจารณาถึงสถานะของสารสนเทศ (State information) ที่จะต้องถูกจดจำอยู่ตลอดเวลา ซึ่งจะช่วยให้สามารถพบลักษณะประจำของวัตถุเพิ่มเติมขึ้นได้

4.1.5) พิจารณาถึงสถานะที่วัตถุสามารถเป็นได้ โดยแต่ละคุณสมบัติเฉพาะจะต้องถูกระบุเพื่อพิจารณาสำหรับแต่ละคุณสมบัติเฉพาะและค่าที่ยอมรับได้

4.2) ความสัมพันธ์ คลาสในระบบจะมีความสัมพันธ์กัน ซึ่งความสัมพันธ์เหล่านี้มีหลายรูปแบบ ดังนี้ [10]

4.2.1) แอสโซซิเอชั่น (Association) เป็นความสัมพันธ์ที่แสดงเส้นทางการเข้าถึง (access path) ระหว่างวัตถุ โดยข้อความบนเส้นใช้อธิบายความสัมพันธ์ การระบุจำนวนตัวแทนที่มีส่วนร่วมในความสัมพันธ์ในแบบจำลองวัตถุจะใช้การระบุตัวเลขที่จุดสิ้นสุดของเส้นที่แสดงความสัมพันธ์

4.2.2) แอกริเกชั่น (Aggregation) เป็นรูปแบบพิเศษของแอสโซซิเอชั่นที่ใช้เพื่อแสดงความสัมพันธ์แบบส่วนประกอบ มีความหมายเหมือนกับความสัมพันธ์แบบโครงร่างทุกส่วนในการวิเคราะห์และออกแบบตามวิธีการของโคดและยอร์ดอน แต่เป็นรูปแบบที่แสดงถึงความสัมพันธ์ที่ขึ้นต่อกัน (Dependency)

4.2.3) การสืบทอดคุณสมบัติ (Inheritance) เป็นความสัมพันธ์ที่มีลักษณะกับความสัมพันธ์แบบโครงร่างทั่วไปเพื่อแสดงการสืบทอดคุณสมบัติของคลาสแม่ไปยังคลาสลูก มีความหมายเหมือนกับความสัมพันธ์แบบโครงร่างทั่วไปในการวิเคราะห์และออกแบบตามวิธีการของโคดและยอร์ดอน

4.2.4) การขึ้นต่อกัน เป็นความสัมพันธ์ที่เกิดจากการเรียกใช้บริการของอีกคลาสหนึ่ง กล่าวคือคลาสของผู้ขอบริการขึ้นอยู่กับบริการของคลาสผู้ให้บริการ แต่ไม่มีความสัมพันธ์กันภายในโครงสร้างของคลาส

5) ระดับกำหนดงานบริการ (Service layer)

5.1) ชนิดของงานบริการโดยทั่วไปมี 2 ชนิดคือ [12]

5.1.1) บริการพื้นฐาน (Basic service) เป็นบริการที่ปกติจะไม่แสดงอยู่ในส่วนแสดงบริการในแผนภาพแสดงคลาส บริการพื้นฐานนี้จะเด่นชัดและคงอยู่เสมอ ใน ทุก ๆ คลาส บริการพื้นฐานนี้คือ การสร้างวัตถุ (Creation) การลบวัตถุ (Removing) การสืบค้น (Searching) การรับและตั้ง

ค่าลักษณะประจำ (Get and set attribute) การเพิ่มและลบการติดต่อของวัตถุ (Add and delete object connection)

5.1.2) บริการที่ขึ้นอยู่กับหน้าที่ ที่ถูกกำหนดโดยขอบเขตของปัญหาเป็นบริการที่จะแสดงอยู่ในส่วนแสดงบริการในแผนภาพแสดงคลาส

5.2) การค้นหาและกำหนดบริการสามารถใช้เทคนิคได้ [12] ดังนี้

5.2.1) การระบุเหตุการณ์เพื่อหาบริการที่สัมพันธ์กับเหตุการณ์นั้น

5.2.2) การระบุสถานะของคลาส เมื่อพิจารณาถึงลักษณะประจำและสถานะของลักษณะประจำอาจจะทำให้พบบริการที่ถูกต้องการเพื่อมาจัดการกับค่าของลักษณะประจำนี้

5.2.3) การระบุข้อความ (Message) ที่ต้องการ เนื่องจากข้อความเป็นการร้องขอให้บริการปฏิบัติงานซึ่งถูกส่งมาจากบริการที่เป็นผู้ส่งไปยังบริการที่เป็นผู้รับ ดังนั้นเมื่อพิจารณาถึงข้อความก็อาจทำให้สามารถระบุถึงบริการเพิ่มเติมได้

5.2.4) การระบุรายละเอียดของบริการ เมื่อพิจารณาถึงรายละเอียดของบริการหนึ่ง ๆ อาจทำให้พบความต้องการที่จะนำบริการอื่นที่จะมาช่วยการทำงานของบริการนี้ซึ่งอาจจะอยู่ในคลาสเดียวกันหรือคลาสอื่นเพื่อให้การทำงานของบริการเป็นผลสำเร็จ

5.3) เทคนิคสำหรับการจัดทำเอกสารและการบรรยายรายละเอียดของบริการมี 3 เทคนิคคือ [12]

5.3.1) แผนภาพซิแนริโอ (Scenerios diagram) เป็นการใช้แผนภาพแสดงลำดับการติดต่อกันของวัตถุเพื่อให้ระบบสารสนเทศทำงานได้เป็นผลสำเร็จ กล่าวคือเป็นการแสดงความสัมพันธ์ของคลาสหนึ่งกับบริการอื่น ๆ จากคลาสเดียวกันหรือต่างคลาสกัน บริการซิแนริโอเปรียบได้กับรูทีนย่อยหรือโพรซีเยอร์ในภาษาโปรแกรมอื่น ๆ หลักการของบริการซิแนริโอ ประกอบด้วย 3 ส่วนคือ คลาส และบริการผู้ส่ง คลาสและบริการผู้รับ พารามิเตอร์และจำนวนบริการที่เรียกใช้ เป็นทางเลือกจะมีหรือไม่มีในแผนภาพก็ได้ การใช้ซิแนริโอก็เพื่อหาวัตถุเพิ่ม ทำให้เข้าใจระบบดีขึ้น สามารถประเมินความผิดพลาดของแบบจำลองวัตถุและทดสอบแบบจำลองวัตถุให้ตรงต่อความต้องการได้

5.3.2) โครงสร้างแบบภาษาอังกฤษ (Structured English) เป็นการบรรยายถึงการทำงานโดยการใช้ภาษาอังกฤษในการบรรยาย ไม่มีแนวทางที่เป็นมาตราฐานแต่สิ่งสำคัญอยู่ที่ความกระทัดรัดและได้ใจความที่สามารถสื่อให้ผู้ใช้เข้าใจและความสามารถปรับเปลี่ยนให้เข้ากับการนำไปใช้กับรูปแบบฮาร์ดแวร์และซอฟต์แวร์ใด ๆ

5.3.3) ตารางการตัดสินใจ (Dicision tables) ซึ่งใช้สำหรับบริการที่มีการตัดสินใจที่ซับซ้อนและเป็นการยากที่จะเขียนบรรยาย ตารางการตัดสินในแบ่งออกเป็น 3 ส่วนคือ เงื่อนไข ซึ่งจะบรรยายถึงเหตุการณ์หรือเงื่อนไขที่มีผลกระทบให้เกิดกฎเกณฑ์ การกระทำ ซึ่งบรรยายถึงการกระทำหรือการตัดสินใจที่เกิดขึ้น และกฎ ซึ่งบรรยายว่าการกระทำจะถูกกระทำเมื่อเงื่อนไขใดเกิดขึ้นบ้าง

2.2.2.2 การออกแบบงานเชิงวัตถุ

จากการทำงานในขั้นตอนของการวิเคราะห์งานเชิงวัตถุพบว่าได้คำนึงถึงส่วนขอบเขตของปัญหาและความรับผิดชอบของระบบงานเป็นสำคัญ เพื่อระบุถึงระบบสารสนเทศที่ต้องการได้เพื่อให้ได้แบบจำลองวัตถุ ต้องทำการทบทวนขอบเขตของปัญหาและทำการพิจารณาถึงวัตถุสำหรับส่วนประกอบอื่น ๆ เพื่อให้ได้ระบบที่ครบถ้วนอีก 3 ส่วนประกอบโดยใช้การทำงานในขั้นตอนการวิเคราะห์เชิงวัตถุคือ

2.2.2.2.1 ส่วนติดต่อกับผู้ใช้ (Human interaction component) [13] ประกอบด้วยวัตถุที่จัดการติดต่อระหว่างวัตถุที่เกี่ยวข้องกับขอบเขตของปัญหาและผู้ใช้ ดังเช่น หน้าต่าง (Windows) และรายงาน ส่วนจัดการข้อมูล (Data management component) ประกอบด้วยวัตถุที่จัดการติดต่อระหว่างวัตถุที่เกี่ยวข้องกับขอบเขตของปัญหาและฐานข้อมูล (Database)

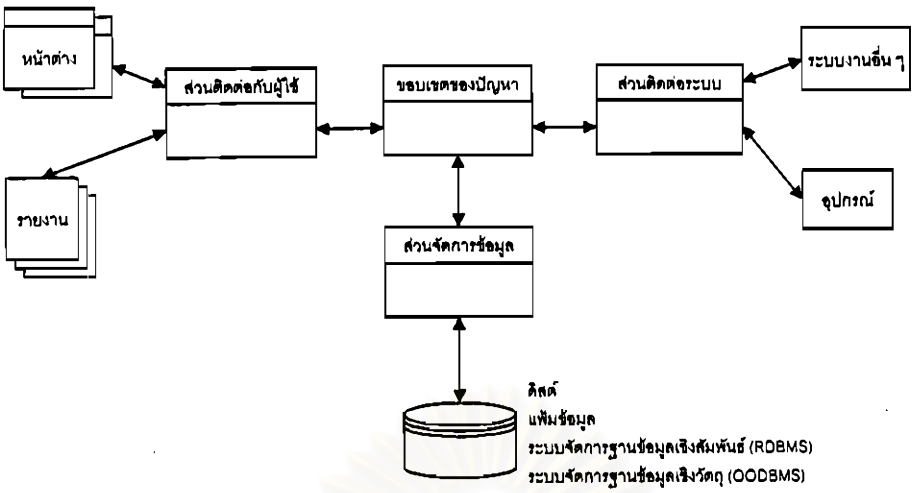
สถาปัตยกรรมของส่วนจัดการข้อมูล (Architecture of Data Management Component) แบ่งเป็น 2 ระดับ (layer) ดังนี้ [11]

1) ระดับตรรกะ (Logical Layer) ประกอบด้วยคลาสที่ทำการแปลงโครงสร้างแบบวัตถุไปสู่ในรูปแบบของความสัมพันธ์

2) ระดับการเข้าถึงฐานข้อมูล (Database Access Layer) เป็นส่วนที่ทำการผนึกส่วนติดต่อกับระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) นั้นคือเป็นระดับที่ประกอบด้วยคลาสที่ทำการประมวลคำสั่งเอสคิวแอล (SQL) เพื่อจัดการตารางของฐานข้อมูล

ในการแบ่งส่วนจัดการข้อมูลออกเป็น 2 ระดับนี้ก่อให้เกิดผลดีในการปรับเปลี่ยน (Portability) การใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์จากระบบหนึ่งไปเป็นอีกระบบหนึ่ง และเป็นการง่ายต่อการนำวัตถุกลับมาใช้ประโยชน์โดยการพัฒนาของระดับการเข้าถึงฐานข้อมูลนี้เอง

2.2.2.2.2 ส่วนติดต่อระบบ (System interaction component) [13] ประกอบด้วยวัตถุที่จัดการติดต่อระหว่างวัตถุที่เกี่ยวข้องกับขอบเขตของปัญหาและระบบอื่น ๆ หรืออุปกรณ์ ยอร์ดอน [8] ได้กล่าวว่า ส่วนติดต่อระบบหรือส่วนในการจัดการ (Task Management Component) นั้นมีความสำคัญต่อการประยุกต์ทำงานแบบทันที (real-time application) ซึ่งมีการทำงานปอย ๆ ในแต่ละวันโดยระบบปฏิบัติการหรือในสิ่งแวดล้อมแบบวิ่ง (run-time evironment) ซึ่งควบคุมรายละเอียดของหลายงาน (multitasking) ให้สามารถทำงานเข้ากันได้ แต่สำหรับการทำงานในสถาปัตยกรรมซอฟต์แวร์ปัจจุบัน ยังไม่มีความต้องการใช้งานส่วนติดต่อระบบอย่างเด่นชัดในการออกแบบ



รูปที่ 2.4 แสดงส่วนประกอบของโครงสร้างเชิงวัตถุและการติดต่อ [13]

2.2.2.3 การพัฒนาโปรแกรมเชิงวัตถุ

วัตถุและความสัมพันธ์ที่ถูกพัฒนา [10] ในระหว่างขั้นตอนการออกแบบจะถูกแปลงโดยภาษาทางโปรแกรม ฐานข้อมูล และการจัดการด้านฮาร์ดแวร์ ซึ่งการเขียนโปรแกรมควรมีความสัมพันธ์กับงานหลักหรือใจความสำคัญที่ได้ทำการออกแบบมา การเขียนโปรแกรมอาจมีการเพิ่มเติมการตัดสินใจบางอย่างได้ แต่ยังคงต้องรักษาใจความสำคัญของการออกแบบไว้ด้วย แนวทางในการพัฒนามี 3 แนวทาง คือ

2.2.2.3.1 ภาษาโปรแกรมเชิงวัตถุทั่วไป (Traditional OO programming languages) [11] เช่น ภาษาซีพลัสพลัส (C++) ภาษาสมอลล์ทอร์ก (Smalltalk) ภาษาไอเฟล (Eiffel) และภาษาอ๊อบเจคปาสคาล (Object Pascal) เป็นต้น

2.2.2.3.2 ภาษาเชิงวัตถุระดับสูงและเครื่องมือ (Higher-level OO languages and tools) [11] เช่น เพาเวอร์บิวเดอร์ (PowerBuilder) และรูปแบบวิชวล (visual) ต่าง ๆ ของภาษาสมอลล์ทอร์ก เป็นต้น

2.2.2.3.3 เครื่องมือช่วยในการพัฒนางานเชิงวัตถุ (OO CASE tools) [11] เช่น Software AG's New,Popkin's System Architect เป็นต้น

ภาษาโปรแกรมเชิงวัตถุทั่วไป [11] ยังคงเป็นแนวทางที่บุคคลส่วนใหญ่ใช้ในการพัฒนาโปรแกรมประยุกต์เชิงวัตถุในช่วงปี ค.ศ. 1990 เนื่องจากความต้องการประสิทธิภาพของโปรแกรมประยุกต์ที่ใหญ่ ซับซ้อน และมีปริมาณรายการเปลี่ยนแปลงจำนวนมาก สำหรับภาษาเชิงวัตถุระดับสูงและเครื่องมือ เป็นอีกทางเลือกหนึ่งที่เหมาะสมสำหรับโปรแกรมประยุกต์ในลักษณะลูกข่าย แม่ข่าย ที่มีขนาดเล็กถึงปานกลาง โดยเฉพาะอย่างยิ่งมีการพัฒนาที่รวดเร็วรวมทั้งการจัดทำโปรแกรมต้นแบบจากส่วนติดต่อผู้ใช้

การใช้เครื่องมือช่วยในการพัฒนางานเชิงวัตถุในการสร้างการวิเคราะห์และออกแบบงานรวมไปถึงการสร้างโปรแกรมนั้นยังไม่แพร่หลาย

2.2.2.4 **การทดสอบระบบ** [8]

ในการทดสอบวัตถุนั้นคล้ายคลึงกับวิธีการที่ใช้ในการทดสอบโปรแกรมเชิงโครงสร้าง คือ การทดสอบระดับหน่วย (unit testing) การทดสอบโดยรวม (integration testing) การทดสอบระบบ (system testing) และการดำเนินการทดสอบสามารถใช้การทดสอบแบบล่างขึ้นบน (bottom - up) หรือการทดสอบแบบบนลงล่าง (top - down)

ในการทดสอบโปรแกรมเชิงวัตถุควรทดสอบโดยรวมแบบบนลงล่าง เนื่องจากโปรแกรมเชิงวัตถุส่วนใหญ่จะเป็นการโปรแกรมที่มีการส่งข้อความ (message-driven) และข้อความส่วนใหญ่จะเริ่มต้นจากการติดต่อระหว่างผู้ใช้กับระบบงาน จึงเป็นการยากที่จะทราบถึงลำดับการประมวลผลและการติดต่อกันระหว่างวัตถุ ทำให้วิธีทดสอบแบบล่างขึ้นบนใช้ไม่ได้ผลดีกับโปรแกรมประยุกต์เชิงวัตถุ

ดังนั้นวิธีทำการทดสอบของโปรแกรมเชิงวัตถุทำให้ต้องพิจารณาการทดสอบแบบเชิงวัตถุได้ดังนี้

2.2.2.4.1 การทดสอบระดับหน่วยเชิงวัตถุ (OO unit testing) เป็นการทดสอบวัตถุแต่ละวัตถุซึ่งเป็นการทดสอบที่ซับซ้อน เนื่องจากความเป็นจริงแล้ววัตถุมักจะมีความเกี่ยวข้องหรือต้องการทำงานร่วมกับวัตถุอื่น ๆ ในระบบ

2.2.2.4.2 การทดสอบโดยรวมเชิงวัตถุ (OO integration testing) เป็นการทดสอบที่เริ่มต้นเมื่อวัตถุเกี่ยวข้องกับวัตถุอื่น ดังนั้นความต้องการหลักของการทดสอบโดยรวมเชิงวัตถุก็คือสิ่งแวดล้อมเชิงวัตถุ (OO environment) ซึ่งก็คือต้องทำการระบุขอบเขตของงานที่ดี

2.2.3 **ภาษาแบบจำลองยูนิฟาย** (Unified Modeling language : UML) [14]

ภาษาแบบจำลองยูนิฟายเป็นวิธีการภาษาแบบจำลองวัตถุในยุคที่ 3 สำหรับการกำหนดและจัดสร้างเอกสารของการพัฒนาระบบงานเชิงวัตถุ เป็นภาษาที่รวบรวมข้อดีของวิธีการของบูช (Booch) อ๊อบเจคทอรี (Objectory) และ โอเอ็มที (OMT) รวมทั้งได้ทำการรวบรวมแนวคิดและวิธีการต่าง ๆ เพิ่มขึ้นด้วย เช่นวิธีการของ ปีเตอร์ โคด (Peter Coad) อีริค แกมม่า (Eric Gamma) เป็นต้น จุดมุ่งหมายของภาษานี้ก็เพื่อให้เป็นภาษาที่มีมาตรฐาน มีความถาวร (stable) และใช้เป็นวิธีการในการพัฒนาระบบเชิงวัตถุ ดังนั้นภาษาแบบจำลองยูนิฟายสามารถประยุกต์ใช้ในการสร้างแบบจำลองระบบทุกรูปแบบ ได้แก่ระบบทำงานแบบทันที (real time system) ระบบลูกข่าย แม่ข่าย (client server system) เป็นต้น

เนื่องจากภาษาแบบจำลองยูนิฟายมุ่งเน้นเพื่อเป็นมาตรฐานของภาษาในการสร้างแบบจำลองวัตถุจึงไม่มีการระบุกระบวนการทำงาน (process) เพื่อสร้างแบบจำลองวัตถุ ดังนั้นงานวิจัยนี้จึงใช้ภาษาแบบจำลองยูนิฟายและใช้กระบวนการทำงานตามการวิเคราะห์และออกแบบงานเชิงวัตถุตามหลักการของโคดและยอร์ดอน

2.2.4 การออกแบบฐานข้อมูล [8]

ในสิ่งแวดล้อมเชิงวัตถุ วิธีการในอุดมคติก็คือการใช้ระบบจัดการฐานข้อมูลเชิงวัตถุ (OODBMS) ซึ่งในกรณีนี้ ทุก ๆ คลาสของคลาสถาวร (persistent class) และทุก ๆ วัตถุของวัตถุถาวรในโครงสร้างซอฟต์แวร์ก็คือวัตถุฐานข้อมูล (database object) ที่ถูกจัดการโดยระบบจัดการฐานข้อมูลเชิงวัตถุ ดังนั้นงานหลักของการทำงานภายใต้สิ่งแวดล้อมแบบนี้ ก็คือการแบ่งแยกระหว่างวัตถุชั่วคราว (transient object) กับวัตถุถาวร (persistent object) แต่ถึงอย่างไรก็ต้องคำนึงถึงการจัดการโครงสร้างคลาส (class hierarchy) ในส่วนขอบเขตของปัญหา (PDC) และส่วนติดต่อกับผู้ใช้ (HIC)

แต่ในปัจจุบันพบว่าสิ่งแวดล้อมในการพัฒนางานด้วยวิธีการเชิงวัตถุนั้นจะต้องมีการติดต่อกับระบบจัดการข้อมูลที่มีความนิยมในการใช้ในปัจจุบันก็คือระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ดังนั้นจึงต้องอาศัยคลาส และวัตถุในส่วนจัดการข้อมูล (DMC) เพื่อทำหน้าที่ติดต่อระหว่างส่วนประกอบต่าง ๆ ของการออกแบบ คือส่วนขอบเขตของปัญหา ส่วนติดต่อกับผู้ใช้ และส่วนติดต่อระบบ (TMCหรือSI) ถ้าหากว่าส่วนติดต่อระบบเหมาะสมกับงาน กับฐานข้อมูล โดยผ่านการส่งข้อความซึ่งคลาสและวัตถุในส่วนจัดการข้อมูลจะทำการอ่านและเขียนข้อมูลระหว่างฐานข้อมูลและวัตถุภายในระบบ โดยทั่วไปกลยุทธ์สำหรับส่วนจัดการข้อมูลของโครงสร้างซอฟต์แวร์ ประกอบด้วย 2 ส่วนคือ

2.2.4.1 การออกแบบ แบบแผนข้อมูล (data layout) ของเขตข้อมูล ระเบียน แถว และตารางบนฐานข้อมูล รายละเอียดของแบบแผนข้อมูลขึ้นอยู่กับโครงสร้างแฟ้มของระบบจัดการข้อมูล (DBMS) เนื่องจากโปรแกรมประยุกต์ในปัจจุบันปฏิบัติงานในสิ่งแวดล้อมของฐานข้อมูลเชิงสัมพันธ์ ซึ่งในกรณีนี้ปกติจะเริ่มต้นด้วยการแปลงคลาสในแบบจำลองวัตถุเข้าสู่รูปแบบของนอร์มอลมาตรฐานอันดับที่ 3 นั้นคือจะกำหนดตารางฐานข้อมูล (database table) ซึ่งบรรยายรายละเอียดการแปลงไว้ในภาคผนวก ค.

2.2.4.2 การออกแบบวิธีการและบริการซึ่งจะถูกใช้เพื่อเข้าถึงข้อมูล เมื่อทำการจัดการข้อมูลที่จะถูกจัดเก็บบนฐานข้อมูลแล้วจะต้องพิจารณาจัดหาวิธีการหรือบริการที่เหมาะสมสำหรับการดึงและปรับปรุงข้อมูล ยิ่งกว่านั้นต้องพิจารณาถึงวิธีการที่จะทำการรวมเข้าไว้ในคลาสและวัตถุสำหรับส่วนขอบเขตปัญหา ซึ่งวิธีการเหล่านี้จะอยู่ในคลาสภายในส่วนจัดการข้อมูล เรียกคลาสเหล่านี้ว่า คลาสผู้จัดเก็บวัตถุ (Object-Saver class)

วิธีที่ใช้ก็คือการจัดหาคลาสในส่วนจัดการข้อมูลให้สำหรับแต่ละคลาสถาวร (persistence class) ในส่วนขอบเขตของปัญหา ส่วนติดต่อกับผู้ใช้และส่วนติดต่อระบบของสถาปัตยกรรมคลาสผู้จัดเก็บวัตถุต้องการที่รู้จักแฟ้มเพื่อทำการเปิด ตำแหน่งที่เหมาะสมของแฟ้ม แถวและสดมภ์ที่เข้าถึง การปรับ

ปรุงเรคคอร์ดหรือตารางด้วยค่าใหม่ แต่ละคลาสผู้จัดเก็บวัตถุนี้จะต้องมีจำนวนของวิธีการซึ่งสามารถถูกเข้าถึงได้โดยผ่านการส่งข้อความจากคลาสที่ติดต่อด้วยในส่วนประกอบอื่น ๆ ของสถาปัตยกรรมซอฟต์แวร์ ดังนี้

2.2.4.2.1 วิธีการที่สามารถสร้างตัวแทนใหม่ของวัตถุถาวรโดยการสร้างระเบียนขึ้นใหม่ภายในตารางเชิงสัมพันธ์ และอื่น ๆ ค่าของลักษณะประจำของวัตถุที่ทำการสร้างขึ้นใหม่นี้จะถูกเริ่มต้นหรือได้ค่ามาจากพารามิเตอร์ของข้อความที่เรียกใช้วิธีการสร้าง (creation method) นี้

2.2.4.2.2 วิธีการที่สามารถทำการอ่านเฉพาะตัวแทนของวัตถุที่ระบุ โดยทำการอ่านระเบียนที่เหมาะสม และทำการประกอบหรือทำการสร้างตัวแทนวัตถุโดยการใส่ในค่าข้อมูล (data value) ที่เหมาะสมในลักษณะประจำ

2.2.4.2.3 วิธีการที่ทำการติดต่อกับคลาสในส่วนขอบเขตของปัญหา ส่วนติดต่อกับผู้ใช้ และส่วนติดต่อระบบโดยผ่านการส่งข้อความเพื่อจัดเก็บตนเอง ข้อความที่ใช้ในการติดต่อควรจะเป็นค่าข้อมูลที่เหมาะสมสำหรับตัวแทนที่ต้องการถูกจัดเก็บ

2.2.5 การนำซอฟต์แวร์เชิงวัตถุกลับมาใช้ประโยชน์ (Object Oriented Software Reuse) [8]

การนำซอฟแวร์กลับมาใช้ประโยชน์เป็นสิ่งที่ทุกคนตระหนักถึงความสำคัญอย่างมาก โดยเฉพาะอย่างยิ่งในยุคปัจจุบันที่มีการแข่งขันทางธุรกิจกันสูง พบว่าองค์การที่มีการประมวลข้อมูลดีที่สุดซึ่งเป็นองค์การที่ประสบความสำเร็จในการนำซอฟต์แวร์กลับมาใช้ประโยชน์ในระดับ 70 ถึง 80 เปอร์เซนต์ ขณะที่องค์การทั่วไปมีการนำซอฟต์แวร์กลับมาใช้ประโยชน์เพียงในระดับ 20 ถึง 30 เปอร์เซนต์เท่านั้น ดังนั้นเมื่อทำการสร้างระบบงานที่ต้องการ 100,000 บรรทัดของโปรแกรม องค์การที่ประสบความสำเร็จในการนำซอฟต์แวร์กลับมาใช้ประโยชน์จะเขียนโปรแกรมใหม่เพียง 20,000 ถึง 30,000 บรรทัด และนำซอฟต์แวร์กลับมาใช้ประโยชน์ถึง 70,000 ถึง 80,000 บรรทัดของโปรแกรมจากไลบรารี (library) ที่จัดเก็บโปรแกรมส่วนที่นำกลับมาใช้ประโยชน์ ในขณะที่หน่วยงานที่ไม่มีการคำนึงถึงการนำซอฟต์แวร์กลับมาใช้ประโยชน์ต้องเขียนโปรแกรมใหม่ถึง 70,000 ถึง 80,000 บรรทัดและนำซอฟต์แวร์กลับมาใช้ประโยชน์เพียง 20,000 ถึง 30,000 บรรทัดจากไลบรารีที่จัดเก็บโปรแกรมส่วนที่นำกลับมาใช้ประโยชน์ เนื่องจากเป็นหน่วยงานที่ไม่มีควบคุมการพัฒนาระบบงานให้ครอบคุมถึงการนำกลับมาใช้ประโยชน์อย่างดี มีใช้เพียงวิธีการนำกลับมาใช้ประโยชน์ทั่ว ๆ ไปคือ การคัดลอกโปรแกรมมาใช้ เป็นต้น

ในอดีตได้มีความพยายามกระทำการนำกลับมาใช้ประโยชน์ โดยมุ่งเน้นไปในการนำโคด (Code) กลับมาใช้ประโยชน์ แต่ปัญหาหลักในการนำโคดกลับมาใช้ประโยชน์ก็คือ การเขียนโปรแกรมเป็นส่วนที่ดำเนินการหลังจากการทำงานที่สำคัญของการพัฒนาระบบงาน นั้นก็คือการวิเคราะห์และออกแบบนั้นเอง ในปัจจุบันองค์การระดับโลกได้ให้ความสำคัญอย่างในการสร้างผลผลิตที่ดีโดยผ่านการนำกลับมาใช้ประโยชน์ใหม่ในระดับการออกแบบ อีกนัยหนึ่งคือในระดับที่มีการกำหนดงานอย่างชัดเจน

ข้อดีในการนำกลับมาใช้ประโยชน์ ประเด็นหลักก็คือการเพิ่มผลผลิต (Productivity) ให้สูงขึ้น ลดเวลาในการทำงาน จากที่กล่าวมาแล้วข้างต้นพบว่า หากหน่วยงานสามารถนำซอฟต์แวร์กลับมาใช้ประโยชน์

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ได้ 80 เปอร์เซนต์ผลผลิตที่ได้จะเป็น 4 เท่าของหน่วยงานที่นำซอฟแวร์กลับมาใช้ประโยชน์เพียง 20 เปอร์เซนต์ ข้อดีที่สำคัญอีกประการของการนำกลับมาใช้ประโยชน์คือ การเพิ่มคุณภาพ (Quality) กล่าวคือส่วนของซอฟแวร์ที่สามารถนำกลับมาใช้ประโยชน์ต้องผ่านการทดสอบ และการควบคุมคุณภาพมาอย่างสูงดังนั้นจึงลดความเสี่ยงต่อผลลัพธ์ที่จะเกิดความผิดพลาดลงในการนำกลับมาใช้ประโยชน์รวมกับการพัฒนาระบบงานใหม่ ข้อดีอีกประการที่มักจะถูกมองข้ามก็คือ การนำกลับมาใช้ประโยชน์ให้วิธีการสำหรับการสร้างต้นแบบ (Prototyping) ต้นแบบที่สร้างจากส่วนที่นำกลับมาใช้ประโยชน์จะให้การทำงานที่ครบถ้วน เนื่องจากมีการทดสอบและการโปรแกรมอย่างครบถ้วน ทำให้สามารถสร้างต้นแบบได้อย่างง่ายและรวดเร็ว

วิธีการในการนำกลับมาใช้ประโยชน์ มี 3 วิธีดังนี้

2.2.5.1 การสืบทอดคุณสมบัติ (Inheritance)

การสืบทอดคุณสมบัติทำให้สามารถกำหนดการทำงานของคลาสหนึ่งในลักษณะของอีกคลาสหนึ่งได้ การนำกลับมาใช้โดยการทำซับคลาส (Subclassing) มักเรียกวิธีการนี้ว่าการนำกลับมาใช้ประโยชน์แบบกล่องขาว (White-Box Reuse) กล่าวคือเป็นการอ้างถึงระดับการมองเห็นของคลาสแม่ด้วยการสืบทอดคุณสมบัตินี้ทำให้รายละเอียดต่าง ๆ ภายในคลาสแม่จะถูกมองเห็นโดยคลาสลูก การสืบทอดคุณสมบัติของคลาสถูกกำหนดไว้อย่างถาวรในเวลาคอมไพล์ (Compile-time) สามารถนำมาใช้งานได้โดยตรงและเป็นวิธีที่ง่ายต่อการปรับปรุงเพื่อการนำกลับมาใช้ประโยชน์ คือคลาสลูกสามารถนำการทำงานของคลาสแม่มาปรับปรุงให้ทำงานเฉพาะขึ้นได้ [15]

แม้ว่าการสืบทอดคุณสมบัติเป็นวิธีที่ง่ายในการนำกลับมาใช้ประโยชน์ แต่วิธีการสืบทอดคุณสมบัติจะเป็นวิธีการที่ดีเมื่อใช้สำหรับขอบเขตพื้นฐานทั่วไป-เฉพาะ (Domain-based generalization-specialization) ที่มีการกำหนดขอบเขตขึ้นอย่างชัดเจน กล่าวคือหากมีการสร้างคลาสทั่วไปโดยแยกปัจจัย (Factor) จากทุก ๆ ลักษณะและบริการทั่วไปโดยไม่มีการกำหนดขอบเขตที่ชัดเจนจะทำให้ได้คลาสที่ไม่สามารถใช้งานได้ และเป็นคลาสที่ทำความเข้าใจได้ยาก และเมื่อมีการสร้างโปรแกรมประยุกต์ใหม่บางครั้งอาจต้องทำการเปลี่ยนแปลงโครงสร้างการสืบทอดคุณสมบัติ โดยการเพิ่มลดลักษณะประจำหรือบริการเพื่อให้ได้คลาสที่เฉพาะงานขึ้น ซึ่งมีวิธีทำ 2 วิธีด้วยกันคือ วิธีที่หนึ่งคือการเปลี่ยนแปลงกับคลาสโดยตรงซึ่งเป็นวิธีที่ก่อให้เกิดผลกระทบกับโปรแกรมประยุกต์อื่น ๆ ที่ใช้คลาสนั้นด้วยทำให้เกิดค่าใช้จ่ายในการงานสูง วิธีที่ 2 ก็คือทำการเปลี่ยนแปลงกับคลาสโดยการเพิ่มเติมลักษณะเฉพาะอะไรก็ตามที่การประยุกต์นั้นต้องการเข้าไปในคลาส นั่นก็คือการทำงานกับขอบเขตงานที่ชัดเจนนั้นเอง อีกประเด็นของการสืบทอดคุณสมบัติที่ต้องคำนึงถึงคือการส่งผลกระทบอย่างต่อเนื่องภายในโครงสร้างการสืบทอดคุณสมบัติ เนื่องจากการเปลี่ยนแปลงคลาสแม่ [13]

โดยสรุปแล้วปัญหาหลักของการนำกลับมาใช้ประโยชน์โดยการสืบทอดคุณสมบัติก็คือการขึ้นต่อกันในการปฏิบัติงานจริง (Inplementation dependency) และขอบเขตของการใช้งาน ดังนั้นการสืบทอดคุณสมบัติสมควรใช้ภายในโปรแกรมประยุกต์เดียวเพื่อที่จะสามารถอยู่ภายใต้ขอบเขตการทำงานที่ชัดเจน

2.2.5.2 ส่วนประกอบวัตถุ (Object Composition หรือ Component)

วิธีการนี้มีแนวคิดว่าวัตถุประกอบด้วยส่วนประกอบอื่น ๆ กล่าวคือวัตถุสามารถทำกิจกรรมใหม่ ๆ ได้โดยการประสานหรือประกอบวัตถุขึ้นเพื่อให้ได้การทำงานที่ซับซ้อนตามต้องการ วิธีส่วนประกอบวัตถุนี้เป็นวิธีที่ต้องอาศัยวัตถุอื่นเพื่อใช้ส่วนติดต่อ (Interface) เพื่อร่วมทำกิจกรรมนั้นให้สำเร็จ มักเรียกวิธีการนี้ว่าการนำกลับมาใช้ประโยชน์แบบกล่องดำ (Black-Box Reuse) กล่าวคือไม่มีรายละเอียดใดของวัตถุที่ถูกนำมาเป็นส่วนประกอบที่สามารถมองเห็นได้ ดังนั้นจึงเป็นวิธีการนำกลับมาใช้ประโยชน์ที่มีการผนึกข้อมูลและบริการเป็นอย่างดี [15]

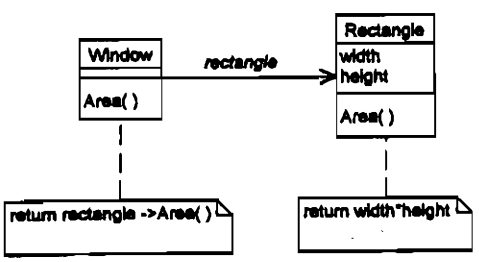
วิธีการนำกลับมาใช้ประโยชน์ด้วยวิธีส่วนประกอบวัตถุนั้น ถูกระบุที่เวลาวิ่งโปรแกรม (Run-time) โดยการอ้างอิงของวัตถุหนึ่งไปยังอีกวัตถุหนึ่ง จากเหตุผลนี้ทำให้สามารถแทนที่วัตถุหนึ่งเป็นอีกวัตถุหนึ่งได้ในขณะวิ่งโปรแกรม ตราบใดที่เป็นวัตถุชนิด (Type) เดียวกัน [15]

การออกแบบระบบงานโดยใช้วิธีการนำกลับมาใช้ประโยชน์วิธีการนี้จะทำให้มีวัตถุจำนวนมากแต่มีจำนวนคลาสน้อย และการทำงานหรือพฤติกรรม (Behavior) ของระบบจะขึ้นกับความสัมพันธ์ของวัตถุมากกว่าวิธีการ (Method) ที่กำหนดขึ้นในคลาสหนึ่ง ๆ [13]

การนำกลับมาใช้ประโยชน์ของด้วยวิธีส่วนประกอบวัตถุนั้น เป็นวิธีการที่เหมาะกับการนำกลับมาใช้ประโยชน์ในงานที่มีหลายโปรแกรมประยุกต์ [13]

ดีลีเกชั่น (Delegation) [15]

ดีลีเกชั่น เป็นแนวทางในการทำส่วนประกอบให้มีการนำกลับมาใช้ประโยชน์อย่างมีการทำงานเท่าเทียบกับการสืบทอดคุณสมบัติ ในการทำดีลีเกชั่นนั้น วัตถุสองวัตถุมีความเกี่ยวข้องกันโดยการส่งการร้องขอ กล่าวคือวัตถุหนึ่งจะทำการมอบหมายให้อีกวัตถุหนึ่งเป็นผู้ปฏิบัติหน้าที่แทน ยกตัวอย่างเช่น แทนที่เราจะสร้างคลาสหน้าต่างโดยการสืบทอดคุณสมบัติจากคลาสสี่เหลี่ยม (Rectangle) เนื่องจากหน้าต่างเป็นสี่เหลี่ยม คลาสหน้าต่างอาจจะนำพฤติกรรม (Behavior) ของคลาสสี่เหลี่ยมกลับมาใช้ประโยชน์โดยการเก็บตัวแปรตัว (Instance) ของคลาสสี่เหลี่ยมไว้แทน และทำการมอบหมายหน้าที่โดยการระบุพฤติกรรมของสี่เหลี่ยมให้กับตัวแปรนี้ ซึ่งในการกระทำเช่นนี้ทำให้หน้าต่างไม่ได้หมายถึงว่าหน้าต่างเป็นสี่เหลี่ยม แต่หมายความว่าหน้าต่างมีลักษณะเป็นสี่เหลี่ยม ซึ่งในขณะนี้หน้าต่างต้องส่งคำร้องขอบริการที่ชัดเจนไปยังตัวแทนของสี่เหลี่ยม



รูปที่ 2.5 แผนภาพแสดงการดีลีเกชั่นของคลาสหน้าต่างไปยังคลาสสี่เหลี่ยม [15]

ข้อดีหลักของการดีลีเกชั่นคือ เป็นวิธีการที่ทำให้ง่ายต่อการประกอบหรือสร้างพฤติกรรมที่เวลาวิ่งโปรแกรมและง่ายต่อการปรับเปลี่ยนแนวทางของพฤติกรรม กล่าวคือหน้าต่างจะกลายมาเป็นวงกลมแทนที่การเป็นสี่เหลี่ยมในขณะรันโปรแกรม ก็โดยการแทนที่ตัวแทน สี่เหลี่ยมด้วยตัวแทนของวงกลม โดยที่สี่เหลี่ยมและวงกลมมีชนิดเดียวกัน

ข้อเสียของการดีลีเกชั่นก็คือมีการใช้เทคนิคอื่น ๆ ร่วมกันเพื่อสร้างซอฟต์แวร์ให้มีความยืดหยุ่นโดยผ่านการทำส่วนประกอบวัตถุ กล่าวคือเป็นวิธีที่เป็นแบบพลวัต (Dynamic) ทำให้ซอฟต์แวร์ใช้พารามิเตอร์จำนวนมากจึงเป็นการยากที่จะทำความเข้าใจมากกว่าซอฟต์แวร์แบบสถิตย์ (Static)

2.2.5.3 การสร้างภาพ (View) [13]

วิธีการนี้เป็นวิธีการที่เหมาะสมกับการนำกลับมาใช้ประโยชน์ในงานที่มีหลายโปรแกรมประยุกต์ กล่าวคือภายใต้แบบจำลองที่ประกอบไปด้วยคลาสและความรับผิดชอบ (Responsibilities) ทั้งหมด แต่ละโปรแกรมประยุกต์ก็คือ การสร้างภาพ การนำกลับมาใช้ประโยชน์วิธีการนี้ มี 2 รูปแบบด้วยกันคือ

2.2.5.3.1 การระบุการใช้งาน (Specialization by usage) วิธีการสร้างภาพแบบนี้มุ่งเน้นในการแสดงถึงความเป็นสามัญ (Commonality) ของคลาสให้ได้มากที่สุด สามารถใช้ได้หลายโปรแกรมประยุกต์โดยการระบุคลาสเฉพาะสำหรับแต่ละการใช้งานในแต่ละโปรแกรมประยุกต์ขึ้น ซึ่งวิธีการระบุการใช้งานนี้ เป็นการนำกลับมาใช้ประโยชน์ที่ใช้ลักษณะการสืบทอดคุณสมบัติ ซึ่งการใช้งานจะต้องคำนึงถึงผลกระทบของการสืบทอดคุณสมบัติด้วย

2.2.5.3.2 การจัดภาพลักษณ์ (View-Based Configuration Management Tool) คือในแต่ละคลาสซึ่งมีความรับผิดชอบจะสามารถเลือกกลุ่มย่อยของความรับผิดชอบขึ้นมาเพื่อให้ได้คลาส ที่ตรงกับความต้องการของโปรแกรมประยุกต์ใหม่ ด้วยวิธีการนี้การนำกลับมาใช้ประโยชน์จะอยู่ในระดับที่สูงกว่าระดับไวยกรณ์ทางภาษาการโปรแกรม แนวทางหนึ่งที่จะทำความเข้าใจเกี่ยวกับการนำกลับมาใช้ประโยชน์วิธีการนี้ก็คือ ทุกๆแบบจำลองขั้นพื้นฐานจะมีแอพลิเคชั่นแทค (Application tag) ซึ่งสัมพันธ์กับการใช้งาน เมื่อเราทำงานภายใต้โปรแกรมประยุกต์ที่กำหนดนั้นก็หมายถึงการทำงานกับแทคพื้นฐานนั้น ด้วยวิธีการนี้เราสามารถรวบรวมหรือประกอบคลาสขึ้นมาอยู่ภายใต้ภาพลักษณ์หรือขอบเขตที่เราต้องการได้

2.2.6 การนำการออกแบบรูปแบบกลับมาใช้ประโยชน์

จากการที่ผู้เชี่ยวชาญด้านเทคโนโลยีเชิงวัตถุได้ทำการออกแบบรูปแบบ แบบจำลองวัตถุ เพื่อให้แบบจำลองวัตถุมีความสามารถในการนำกลับไปใช้ประโยชน์ได้ และได้ทำการแก้ไขปัญหาเบื้องต้นทุก ๆ ปัญหาจึงทำให้รูปแบบที่ได้สามารถแก้ปัญหาเฉพาะของรูปแบบนั้น ๆ ได้และทำให้มีการออกแบบเชิงวัตถุที่ยืดหยุ่น สามารถช่วยให้ผู้ออกแบบประสบผลสำเร็จในการนำการออกแบบกลับไปใช้ประโยชน์และสามารถประยุกต์ใช้ได้กับปัญหาที่ตรงกับปัญหาที่รูปแบบได้มีการแก้ไขมาแล้ว [15]

ดังนั้นจึงนำการออกแบบรูปแบบมาใช้เป็นแม่แบบของความรับผิดชอบและความสัมพันธ์ของวัตถุ ในขอบเขตงานที่กำลังออกแบบก็เพื่อให้เกิดความยืดหยุ่นและการนำกลับไปใช้ประโยชน์ได้ของการออกแบบ นอกจากนี้ยังอาจจะทำให้การออกแบบงานค้นพบวัตถุใหม่เพิ่มเติมเนื่องจากการนำรูปแบบมาออกแบบงานอีกด้วย

สามารถแบ่งรูปแบบที่เกี่ยวข้องในงานวิจัยได้ดังนี้

2.2.6.1 รูปแบบด้านรายการเปลี่ยนแปลง (Transaction Patterns) [13] ซึ่งงานวิจัยได้นำไปใช้เป็นแม่แบบของกลุ่มวัตถุที่เกี่ยวข้องกับรายการเปลี่ยนแปลงที่เกิดขึ้นในกลุ่มงานต่าง ๆ รูปแบบนี้ประกอบด้วยรูปแบบดังนี้

2.2.6.1.1 รูปแบบนักแสดง - ผู้มีส่วนร่วม (Actor - Participant) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างนักแสดงและผู้มีส่วนร่วมซึ่งเป็นบทบาทของนักแสดง โดยมีความสัมพันธ์ในลักษณะหนึ่งต่อหลาย กล่าวคือวัตถุนักแสดง 1 คนรู้จักผู้มีส่วนร่วมหลายคน

ก. ตัวอย่างนักแสดง เช่น บุคคล องค์การ เป็นต้น

ข. ตัวอย่างผู้มีส่วนร่วม เช่น ผู้สมัคร ผู้ขาย คนเก็บเงิน ลูกค้า เป็นต้น

2.2.6.1.2 รูปแบบผู้มีส่วนร่วม - รายการเปลี่ยนแปลง (Participant - Transaction Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างผู้มีส่วนร่วมและรายการเปลี่ยนแปลงที่เกิดขึ้น โดยมีความสัมพันธ์ในลักษณะหนึ่งต่อหลาย กล่าวคือวัตถุผู้มีส่วนร่วมหนึ่งรู้จักวัตถุรายการเปลี่ยนแปลงหลายรายการ

ก. ตัวอย่างผู้มีส่วนร่วม เช่น ผู้สมัคร ผู้ขาย คนเก็บเงิน ลูกค้า เป็นต้น

ข. ตัวอย่างรายการเปลี่ยนแปลง เช่น ข้อตกลง สัญญา การนำฝาก การสั่งการจ่าย เป็นต้น

2.2.6.1.3 รูปแบบสถานที่ - รายการเปลี่ยนแปลง (Place - Transaction) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างสถานที่และรายการเปลี่ยนแปลงที่สัมพันธ์กับสถานที่นี้ โดยมีลักษณะความสัมพันธ์แบบหนึ่งต่อหลาย กล่าวคือวัตถุสถานที่ 1 แห่งรู้จักวัตถุรายการเปลี่ยนแปลงหลายรายการ

ก. ตัวอย่างสถานที่ เช่น สนามบิน ธนาคาร จุดขาย หน่วยบริการ คลังสินค้า เป็นต้น

ข. ตัวอย่างรายการเปลี่ยนแปลง เช่น ข้อตกลง สัญญา การนำฝาก การสั่งการจ่าย เป็นต้น

2.2.6.1.4 รูปแบบพัสดุที่กำหนด - รายการเปลี่ยนแปลง (Specific Item - Transaction Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างพัสดุที่กำหนดและรายการเปลี่ยนแปลง โดยมีความสัมพันธ์ในลักษณะหนึ่งต่อหลาย กล่าวคือวัตถุพัสดุที่กำหนดหนึ่งหน่วยรู้จักวัตถุรายการเปลี่ยนแปลงหลายรายการ

ก. ตัวอย่างพัสดุที่กำหนด เช่น เครื่องบินที่กำหนด เรือที่กำหนด เป็นต้น

ข. ตัวอย่างรายการเปลี่ยนแปลง เช่น ข้อตกลง สัญญา การนำฝาก การสั่งการจ่าย เป็นต้น

2.2.6.1.5 รูปแบบรายการเปลี่ยนแปลง - รายละเอียดรายการเปลี่ยนแปลง (Transaction - Transaction Line Item Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างรายการเปลี่ยนแปลงและรายละเอียดรายการเปลี่ยนแปลงนี้ โดยมีความสัมพันธ์ในลักษณะโครงร่างทุกส่วน กล่าวคือ วัตถุรายการเปลี่ยนแปลง 1 รายการประกอบด้วยวัตถุรายละเอียดรายการเปลี่ยนแปลงหลายรายการ

ก. ตัวอย่างรายการเปลี่ยนแปลง เช่น ข้อตกลง สัญญา การนำฝาก การสั่ง การจ่าย เป็นต้น

ข. ตัวอย่างรายการเปลี่ยนแปลง - รายละเอียดรายการเปลี่ยนแปลง เช่น การนำฝาก - รายละเอียดการนำฝาก การสั่ง - รายละเอียดการสั่ง การจ่าย - รายละเอียดการจ่าย

2.2.6.1.6 รูปแบบรายการเปลี่ยนแปลง - ผลต่อเนื่องรายการเปลี่ยนแปลง (Transaction - Subsequent Transaction Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างรายการเปลี่ยนแปลงและผลต่อเนื่องรายการเปลี่ยนแปลงที่เกิดจากรายการเปลี่ยนแปลงนี้ โดยมีความสัมพันธ์ในลักษณะหนึ่งต่อหลาย กล่าวคือวัตถุรายการเปลี่ยนแปลง 1 รายการรู้จักวัตถุผลต่อเนื่องรายการเปลี่ยนแปลงหลายรายการ

ก. ตัวอย่างรายการเปลี่ยนแปลง เช่น ข้อตกลง สัญญา การนำฝาก การสั่ง การจ่าย เป็นต้น

ข. ตัวอย่างรายการเปลี่ยนแปลง - ผลต่อเนื่องรายการเปลี่ยนแปลง เช่น การสั่ง - การส่งของ การซื้อ - การจ่าย การจอง - การขาย เป็นต้น

2.2.6.1.7 รูปแบบรายละเอียดรายการเปลี่ยนแปลง - รายละเอียดผลต่อเนื่องรายการเปลี่ยนแปลง (Transaction Line Item - Subsequent Transaction Line Item Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างรายละเอียดรายการเปลี่ยนแปลง - รายละเอียดผลต่อเนื่องรายการเปลี่ยนแปลงที่เกิดจากรายละเอียดรายการเปลี่ยนแปลงนี้ โดยมีความสัมพันธ์ในลักษณะหนึ่งต่อหลาย กล่าวคือวัตถุรายละเอียดรายการเปลี่ยนแปลง 1 รายการรู้จักวัตถุรายละเอียดผลต่อเนื่องรายการเปลี่ยนแปลงหลายรายการ

ก. ตัวอย่างรายละเอียดรายการเปลี่ยนแปลง เช่น ข้อตกลง สัญญา การนำฝาก การสั่ง การจ่าย เป็นต้น

ข. ตัวอย่างรายละเอียดรายการเปลี่ยนแปลง - รายละเอียดผลต่อเนื่องรายการเปลี่ยนแปลง เช่น รายละเอียดการสั่ง - รายละเอียดการส่ง รายละเอียดการจอง - รายละเอียดการเช่า เป็นต้น

2.2.6.1.8 รูปแบบพัสดุ - รายละเอียดรายการ (Item - Line Item Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างพัสดุและรายละเอียดรายการ โดยมีความสัมพันธ์ในลักษณะหนึ่งต่อหลาย

ก. ตัวอย่างพัสดุ - รายละเอียดรายการเปลี่ยนแปลง เช่น พัสดุ - รายละเอียดการสั่ง พัสดุ - รายละเอียดการจ่าย พัสดุ - รายละเอียดการเช่า เป็นต้น

ข. ตัวอย่างพัสดุ - รายละเอียดรายการสิ่งบรรจุ เช่น พัสดุ - รายละเอียดรายการคลังพัสดุ เป็นต้น

2.2.6.1.9 รูปแบบพัสดุที่กำหนด- รายละเอียดรายการ (Specific Item - Line Item Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างพัสดุที่กำหนดและรายละเอียดรายการ โดยมีความสัมพันธ์ในลักษณะหนึ่งต่อหลาย กล่าวคือวัตถุพัสดุที่กำหนด 1 หน่วยรู้จักวัตถุรายละเอียดรายการหลายรายการ

ตัวอย่างพัสดุที่กำหนด- รายละเอียดรายการ เช่น เครื่องบินที่กำหนดเฉพาะ - รายละเอียดรายการ เรือที่กำหนดเฉพาะ - รายละเอียดรายการ เป็นต้น

2.2.6.1.10 รูปแบบพัสดุ - พัสดุที่กำหนด (Item - Specific Item Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างพัสดุและพัสดุที่กำหนดซึ่งจะสามารถระบุชนิดของพัสดุได้ โดยมีความสัมพันธ์ในลักษณะหนึ่งต่อหลาย กล่าวคือวัตถุพัสดุ 1 หน่วยรู้จักวัตถุพัสดุที่กำหนดหลายหน่วย

ตัวอย่างพัสดุ - พัสดุที่กำหนด คือ เครื่องบิน - เครื่องบินที่กำหนด คำบรรยายวีดีโอ - วีดีโอเทป เป็นต้น

2.2.6.1.11 รูปแบบความแอสโซซิเอท - แอสโซซิเอทอื่น (Associate - Other Associate) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างความสัมพันธ์และความสัมพันธ์อื่น โดยมีความสัมพันธ์ในลักษณะหนึ่งต่อหลาย กล่าวคือวัตถุความสัมพันธ์หนึ่งรู้จักวัตถุความสัมพันธ์อื่นหลายความสัมพันธ์

ตัวอย่างความสัมพันธ์ - ความสัมพันธ์อื่น เช่น เครื่องบิน - ทางวิ่ง เป็นต้น

2.2.6.1.12 รูปแบบพัสดุที่กำหนด - โครงสร้างพัสดุ (Specific Item - Hierarchical Item) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างพัสดุที่กำหนดและโครงสร้างพัสดุของพัสดุนี้ โดยมีความสัมพันธ์ในลักษณะทางเดียวคือวัตถุพัสดุที่กำหนด 1 หน่วยรู้จักโครงสร้างพัสดุหลายพัสดุในโครงนั้น

ตัวอย่างพัสดุที่กำหนด - โครงสร้างพัสดุ เช่น บัญชี - โครงสร้างบรรยายบัญชี องค์การ - โครงสร้างบรรยายองค์การ เป็นต้น

2.2.6.2 รูปแบบแอกริเกท (Aggregate Patterns) [13] ซึ่งงานวิจัยได้นำไปใช้เพื่อเป็นแม่แบบแก่กลุ่มวัตถุที่มีลักษณะของส่วนประกอบในกลุ่มงานย่อยต่าง ๆ รูปแบบนี้ประกอบด้วยรูปแบบดังนี้

2.2.6.2.1 รูปแบบสิ่งบรรจุ - สิ่งที่บรรจุภายใน (Container - Content Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างสิ่งบรรจุและสิ่งที่บรรจุภายใน โดยมีความสัมพันธ์ในลักษณะโครงร่างทุกส่วน กล่าวคือวัตถุสิ่งบรรจุ 1 หน่วยบรรจุด้วยวัตถุที่เป็นสิ่งบรรจุภายในหลายหน่วย

ก. ตัวอย่างสิ่งบรรจุ เช่น สนามบิน เครื่องบิน ธนาคาร คลังสินค้า ที่เก็บของ เป็นต้น

ข. ตัวอย่างสิ่งบรรจุ - สิ่งที่บรรจุภายใน เช่น เครื่องบิน - สินค้า ตึก - ห้อง เป็นต้น

2.2.6.2.2 รูปแบบสิ่งบรรจุ - รายละเอียดสิ่งบรรจุ (Container - Container Line Item Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างสิ่งบรรจุและรายละเอียดสิ่งบรรจุ โดยมีความสัมพันธ์ในลักษณะโครงร่างทุกส่วน กล่าวคือวัตถุสิ่งบรรจุ 1 หน่วยมีรายละเอียดสิ่งบรรจุหลายรายการ

ตัวอย่างสิ่งบรรจุ - รายละเอียดสิ่งบรรจุ เช่น เครื่องบิน - รายละเอียดเครื่องบิน คลังสินค้า - รายละเอียดคลังสินค้า เป็นต้น

2.2.6.2.3 รูปแบบกลุ่ม - สมาชิก (Group - Member Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างกลุ่มและสมาชิก โดยมีความสัมพันธ์ในลักษณะโครงร่างทุกส่วน กล่าวคือ วัตถุกลุ่ม 1 กลุ่มมีสมาชิกหลายหน่วย

ตัวอย่างกลุ่ม - สมาชิก เช่น บริษัท - พนักงาน ทีม - สมาชิกทีม เป็นต้น

2.2.6.2.4 รูปแบบการรวบรวม - ส่วนประกอบ (Assembly - Part Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างการรวบรวมและส่วนประกอบ โดยมีความสัมพันธ์ในลักษณะโครงร่างทุกส่วน กล่าวคือวัตถุการรวบรวมหนึ่งประกอบด้วยส่วนประกอบหลายส่วน

ตัวอย่างการรวบรวม - ส่วนประกอบ เช่น เครื่องบิน - เครื่องยนต์ เครื่องยนต์ - ส่วนประกอบเครื่องยนต์ เป็นต้น

2.2.6.2.5 รูปแบบการประกอบส่วนประกอบ - ส่วนประกอบ (Compound Part - Part Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างการประกอบส่วนประกอบและส่วนประกอบ โดยมีความสัมพันธ์ในลักษณะโครงร่างทุกส่วน กล่าวคือวัตถุการประกอบส่วนประกอบหนึ่งมีส่วนประกอบหลายส่วน

ตัวอย่างการประกอบส่วนประกอบ - ส่วนประกอบ เช่น การประกอบส่วนอิเล็คทรอนิกส์ - ส่วนอิเล็คทรอนิกส์ เป็นต้น

2.2.6.2.6 รูปแบบการบรรจุหีบห่อ - ส่วนบรรจุ (Packet - Packet Component Pattern) เป็นรูปแบบของลักษณะประจำและการติดต่อระหว่างการบรรจุหีบห่อและส่วนบรรจุ โดยมีความสัมพันธ์ในลักษณะโครงร่างทุกส่วน กล่าวคือวัตถุการบรรจุหีบห่อหนึ่งประกอบด้วยวัตถุส่วนบรรจุหลายส

ตัวอย่างการบรรจุหีบห่อ - ส่วนบรรจุ เช่น ระเบียน - เขตข้อมูล เป็นต้น

2.2.6.3 รูปแบบลักษณะภายนอก (Facade) [15] เป็นรูปแบบที่ทำการรวบรวมส่วนติดต่อเพื่อให้เป็นกลุ่มของส่วนติดต่อในระบบงานย่อย รูปแบบลักษณะภายนอกนี้จะกำหนดส่วนติดต่อในระดับที่สูง (higher-level) ก็เพื่อให้ระบบงานย่อยง่ายต่อการติดต่อ วัตถุที่เป็นลักษณะภายนอกนี้จะรู้จักคลาสในระบบงานย่อยที่สามารถให้บริการที่ต้องการได้ และจะทำการดีลีเกทงานไปยังวัตถุที่เหมาะสมในระบบงานย่อยนี้

2.2.6.4 รูปแบบสถานะ (State) [15] เป็นรูปแบบที่ทำให้วัตถุเปลี่ยนแปลงพฤติกรรมได้เมื่อสถานะภายในมีการเปลี่ยนแปลง และมีเงื่อนไขการทำงานที่ขึ้นกับสถานะของวัตถุ โดยปกติสถานะจะแสดงด้วยการระบุค่าคงที่หนึ่งหรือหลายจำนวน โดยการทำงานหลาย ๆ ส่วนมีโครงสร้างเงื่อนไขเหมือนกัน ดังนั้นรูปแบบสถานะจึงแยกแต่ละส่วน (branch) เงื่อนไขไปอยู่ในแต่ละคลาสที่แสดงแต่ละสถานะ งานวิจัยได้นำรูปแบบนี้ไปใช้ในกลุ่มงานทางธุรกิจเพื่อให้วัตถุสามารถปรับเปลี่ยนสถานะได้ง่ายและนำไปใช้กับวัตถุที่มีการทำงานและสร้างความสัมพันธ์กับวัตถุอื่นเปลี่ยนแปลงไปขึ้นกับสถานะของวัตถุ เช่น ในกลุ่มงานต้นทุนการผลิตเพื่อเป็นสถานะของงานระหว่างทำ เนื่องจากวิธีการคำนวณต้นทุนผลิตภัณฑ์ซึ่งมีเงื่อนไขในการคำนวณที่จำเป็นต้องคำนึงถึงงานระหว่างทำปัจจุบันและงานระหว่างทำต้นงวดการผลิต ในกลุ่มงานรับเงินเพื่อให้คลาสใบสำคัญรับเงินมีสถานะการรับเงินได้หลายสถานะมีความสามารถในการปรับเปลี่ยนสถานะได้ง่ายและเพื่อให้คลาสใบสำคัญรับเงินมีความสัมพันธ์และวิธีการทำงานขึ้นกับสถานะ เป็นต้น

2.2.6.5 รูปแบบกลวิธี (Startegy) [15] เป็นรูปแบบที่ทำการกำหนดกลุ่มของอัลกอริทึม (algorithms) ทำการผนึกแต่ละอัลกอริทึมและสามารถแลกเปลี่ยนการทำงานได้ รูปแบบกลวิธีจะทำให้

อัลกอริทึมเป็นอิสระจากวัตถุที่เรียกใช้บริการ งานวิจัยได้นำรูปแบบนี้ไปเป็นแม่แบบในกลุ่มงานที่มีการใช้หลายกลวิธีหรือหลายวิธีการในการคำนวณของงานนั้น ๆ เช่นงานสินทรัพย์ถาวรในส่วนการคำนวณค่าเสื่อมราคา

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 วิทยานิพนธ์มหาบัณฑิตเรื่อง การศึกษาข้อมูลเพื่อเป็นแนวทางการวางระบบงานที่ใช้คอมพิวเตอร์สำหรับธุรกิจผลิตและจำหน่าย พ.ศ. 2525 จัดโดย น.ส. วิมลพร ชลาเธิงพันธ์ ซึ่งเป็นวิทยานิพนธ์ตามหลักสูตรปริญญาวิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ได้ทำการศึกษาข้อมูลเพื่อเป็นแนวทางการวางระบบงานที่ใช้คอมพิวเตอร์สำหรับธุรกิจการผลิตและจำหน่ายคือ วิธีการนำคอมพิวเตอร์มาใช้ในธุรกิจนั้น ควรจะเริ่มพิจารณาข้อมูลส่วนใดก่อนจึงเหมาะสม การจัดวางระบบหน่วยงานของกิจการ การจัดวางรหัสของหน่วยงานเพื่อใช้เป็นตัวกลางในการดำเนินงานด้านคอมพิวเตอร์ [16]

2.3.2 วิทยานิพนธ์มหาบัณฑิตเรื่อง การพัฒนาระบบต้นทุนการผลิตของโรงงานผลไม้อบแห้งด้วยคอมพิวเตอร์ ปี พ.ศ. 2539 จัดโดย นายวีระชัย จันจงเจริญชัย ซึ่งเป็นวิทยานิพนธ์ตามหลักสูตรปริญญาวิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ได้ทำการวิเคราะห์ ออกแบบ และพัฒนาระบบต้นทุนนี้ โดยศึกษาองค์ประกอบหลักของระบบการคำนวณต้นทุนได้แก่ การผลิต ลักษณะต้นทุน ต้นทุนวัตถุดิบ ต้นทุนแรงงาน ต้นทุนค่าใช้จ่ายการผลิต รวมทั้งการคำนวณต้นทุนงานระหว่างทำ ผลิตภัณฑ์สำเร็จรูป และปัจจัยต่าง ๆ [17]

2.3.3 โครงงานพิเศษเรื่อง ระบบสารสนเทศทางการบัญชีการเงินของธุรกิจผลิตน้ำดื่มบริสุทธิบรรจุขวด ปี พ.ศ. 2538 จัดทำโดย น.ส. รัชนี ชุมขุนทด ซึ่งโครงงานพิเศษตามหลักสูตรปริญาวิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศทางธุรกิจ แขนงวิชาระบบสารสนเทศเพื่อการบัญชี คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ได้ทำการวิเคราะห์ ออกแบบและพัฒนาระบบบัญชีการเงิน โดยคำนึงถึงการควบคุมภายในที่รัดกุมและคล่องตัวในการปฏิบัติงาน และการเชื่อมโยงกับระบบสารสนเทศที่เกี่ยวข้อง [18]

2.3.4 โครงงานพิเศษเรื่อง ระบบสารสนเทศทางการบัญชีต้นทุนของธุรกิจผลิตน้ำดื่มบริสุทธิบรรจุขวด ปี พ.ศ. 2538 จัดทำโดย นายไปรม์ ปานะนนท์ ซึ่งโครงงานพิเศษตามหลักสูตรปริญาวิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศทางธุรกิจ แขนงวิชาระบบสารสนเทศเพื่อการบัญชี คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ได้ทำการวิเคราะห์ ออกแบบ และพัฒนาระบบงานบัญชีต้นทุน โดยทำการกำหนดต้นทุนโดยประมาณของสินค้า ทั้งในส่วนที่เป็นวัตถุดิบและค่าใช้จ่ายการผลิต การคำนวณต้นทุนโดยประมาณที่เกิดขึ้นในแต่ละงวด การรวบรวมต้นทุนจริงที่เกิดขึ้นในแต่ละงวด การคำนวณหาผลต่างการผลิตในแต่ละงวด การจัดทำงบต้นทุนการผลิต [19]