สำนักงานวิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
การแปลงแผนภาพคลาสและแผนภาพซีเควนซ์เป็นแอ็บสแตร็คแมชชีนบี

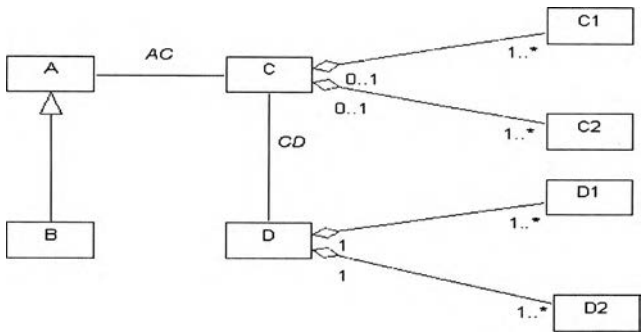
ในบทนี้จะกล่าวถึงการแปลงแผนภาพคลาสและแผนภาพซีเควนซ์เป็นแอ็บสแตร็คแมชชีนบี โดยการนำเสนอกฎการแปลง ซึ่งประกอบด้วย กฎการสร้างแอ็บสแตร็คแมชชีนบีจากแผนภาพคลาส และกฎการสร้างแอ็บสแตร็คแมชชีนบีที่มีโอเปอเรชันคลาสที่แสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คของหลาย ๆ เหตุการณ์จากแผนภาพซีเควนซ์ โดยกฎการสร้างแอ็บสแตร็คแมชชีนบีจากแผนภาพคลาส ประกอบด้วยการสร้างแอ็บสแตร็คแมชชีนบีของคลาส การสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชัน ความสัมพันธ์แอกริเกชัน และความสัมพันธ์คอมโพสิชันของคลาส การสร้างแอ็บสแตร็คแมชชีนบีซับคลาส รวมทั้งการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์โดยปริยายระหว่างซับคลาสกับคลาสอื่น ๆ จากความสัมพันธ์เจเนอรัลไลเซชันของคลาส ส่วนกฎการสร้างแอ็บสแตร็คแมชชีนบีจากแผนภาพซีเควนซ์ จะเป็นการสร้างแอ็บสแตร็คแมชชีนบีที่มีโอเปอเรชันคลาสจากรูปแบบการรับส่งข้อความระหว่างออบเจ็คของหลาย ๆ เหตุการณ์จากแผนภาพซีเควนซ์ และทำการสร้างอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีเพื่อแสดงการเรียกโอเปอเรชันคลาสที่แสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คของหลาย ๆ เหตุการณ์จากแผนภาพซีเควนซ์

3.1 คลาส และความสัมพันธ์ระหว่างคลาสในแผนภาพคลาสกับกฎการแปลง

จากทฤษฎีที่เกี่ยวข้องกับคลาส และความสัมพันธ์ระหว่างคลาสในแผนภาพคลาส ทำการสร้างแอ็บสแตร็คแมชชีนบีของคลาส ซึ่งอธิบายในหัวข้อที่ 3.2.1 ประกอบด้วย กฎข้อที่ 1 กฎข้อที่ 2 และกฎข้อที่ 3 ส่วนการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์ได้แก่ ความสัมพันธ์แอสโซซิเอชัน ความสัมพันธ์แอกริเกชัน และความสัมพันธ์คอมโพสิชันของคลาส จะอธิบายในหัวข้อที่ 3.2.2 ประกอบด้วย กฎข้อที่ 4 กฎข้อที่ 5 และกฎข้อที่ 6 การสร้างแอ็บสแตร็คแมชชีนบีซับคลาส และการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์โดยปริยายระหว่างซับคลาสกับคลาสอื่น ๆ จากความสัมพันธ์เจเนอรัลไลเซชันของคลาส จะอธิบายในหัวข้อที่ 3.2.3 ประกอบด้วย กฎข้อที่ 7 และกฎข้อที่ 8

คลาส และความสัมพันธ์ที่มีอยู่ในแผนภาพคลาส จะแสดงได้ดังรูปที่ 3.1

หอสมุดกลาง สำนักงานวิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.1 คลาส และความสัมพันธ์ในแผนภาพคลาส

ในงานวิจัยนี้เป็นการนำเสนอกฎการแปลงคลาส และความสัมพันธ์ในแผนภาพคลาส จากรูปที่ 3.1 จะพบว่าคลาสในแผนภาพคลาส ได้แก่ ซูเปอร์คลาส A ซับคลาส B คลาส C คลาส C1 คลาส C2 คลาส D คลาส D1 และคลาส D2 รวมทั้งความสัมพันธ์ทั้งหมดในแผนภาพคลาส ได้แก่ ความสัมพันธ์แอสซิเอชัน ได้แก่ ความสัมพันธ์ AC และ CD ความสัมพันธ์เจเนอรัลไลเซชันของซูเปอร์คลาส A กับซับคลาส B ความสัมพันธ์แอกริเกชันของคลาส C กับคลาส C1 และคลาส C2 และความสัมพันธ์คอมโพสิชันของคลาส D กับคลาส D1 และคลาส D2

จากรูปที่ 3.1 นี้ จะแสดงแอ็บสแตร็คแมชชีนบีของคลาส และความสัมพันธ์ทั้งหมดในแผนภาพคลาสที่ได้ทำการออกแบบ มีดังนี้

1) แอ็บสแตร็คแมชชีนบีซูเปอร์คลาส A
2) แอ็บสแตร็คแมชชีนบีเบซิคซูเปอร์คลาส A
3) แอ็บสแตร็คแมชชีนบีซับคลาส B
4) แอ็บสแตร็คแมชชีนบีเบซิคซับคลาส B
5) แอ็บสแตร็คแมชชีนบีคลาส C
6) แอ็บสแตร็คแมชชีนบีเบซิคคลาส C
7) แอ็บสแตร็คแมชชีนบีคลาส C1
8) แอ็บสแตร็คแมชชีนบีเบซิคคลาส C1
9) แอ็บสแตร็คแมชชีนบีคลาส C2
10) แอ็บสแตร็คแมชชีนบีเบซิคคลาส C2
11) แอ็บสแตร็คแมชชีนบีคลาส D
12) แอ็บสแตร็คแมชชีนบีเบซิคคลาส D
13) แอ็บสแตร็คแมชชีนบีคลาส D1
14) แอ็บสแตร็คแมชชีนบีเบซิคคลาส D1
15) แอ็บสแตร็คแมชชีนบีคลาส D2

16) แอ็บสแตร็คแมชชีนบีเบซิคคลาส D2

17) แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาส A กับคลาส C

18) แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาส C กับคลาส D

19) แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่างซับคลาส B กับคลาส C

20) แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาส C กับคลาส C1

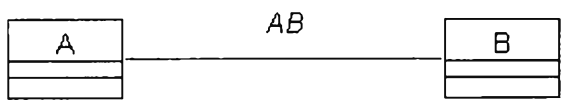
21) แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาส C กับคลาส C2

22) แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันชันระหว่างคลาส D กับคลาส D1

23) แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันชันระหว่างคลาส D กับคลาส D2

3.2 กฎการสร้างแอ็บสแตร็คแมชชีนบีของคลาส และความสัมพันธ์จากแผนภาพคลาส

ในการสร้างแอ็บสแตร็คแมชชีนบีจากแผนภาพคลาส จะประกอบไปด้วย ส่วนของการสร้างแอ็บสแตร็คแมชชีนบีของคลาส และส่วนของการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์ระหว่างคลาส โดยพิจารณาแผนภาพคลาสได้ดังรูปที่ 3.2



รูปที่ 3.2 แผนภาพคลาส ซึ่งประกอบไปด้วย คลาส A คลาส B และความสัมพันธ์ AB

จากรูปที่ 3.2 อาศัยกฎการสร้างแอ็บสแตร็คแมชชีนบีจากแผนภาพคลาส จะได้แอ็บสแตร็คแมชชีนบีทั้งหมด ดังนี้

1) แอ็บสแตร็คแมชชีนบีคลาส A

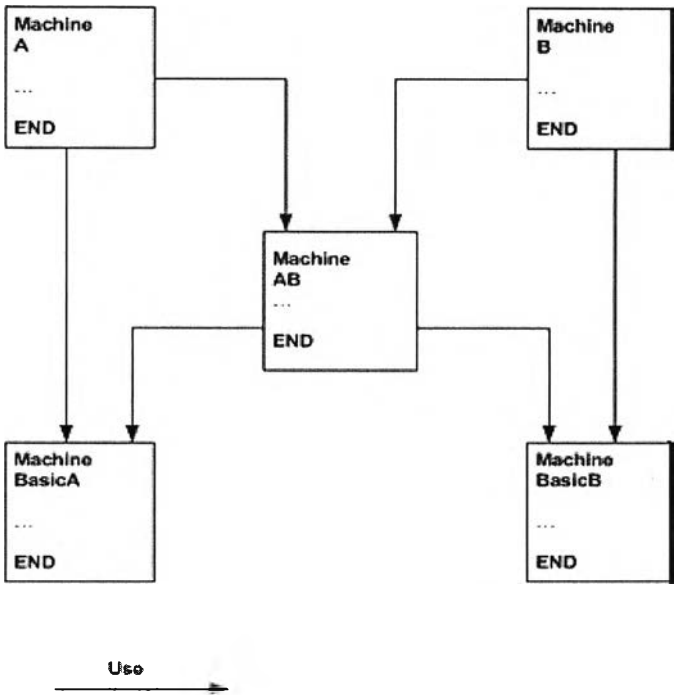
2) แอ็บสแตร็คแมชชีนบีเบซิคคลาส A

3) แอ็บสแตร็คแมชชีนบีคลาส B

4) แอ็บสแตร็คแมชชีนบีเบซิคคลาส B

5) แอ็บสแตร็คแมชชีนบีของความสัมพันธ์ระหว่างคลาส A กับคลาส B

แสดงรูปแบบของสถาปัตยกรรมแอ็บสแตร็คแมชชีนบี (B Abstract Machine Architecture) ได้ดังรูปที่ 3.3

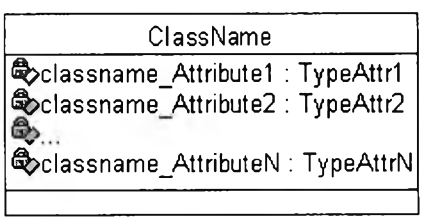


รูปที่ 3.3 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีจากคลาส A คลาส B และความสัมพันธ์ AB

3.2.1 การสร้างแอ็บสแตร็คแมชชีนบีของคลาส

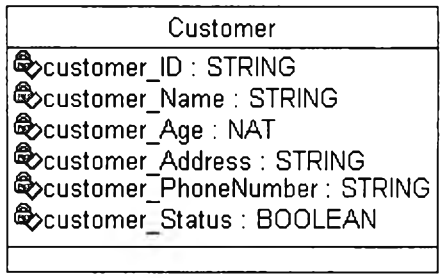
ในการสร้างแอ็บสแตร็คแมชชีนบีของคลาสที่แสดงถึงชื่อของคลาส ชื่อของคุณลักษณะ และชนิดของคุณลักษณะทั้งหมดของคลาสจากแผนภาพคลาส ซึ่งประกอบด้วยชื่อของคลาส ClassName ชื่อของคุณลักษณะ ได้แก่ classname_Attribute1, classname_Attribute2, ..., classname_AttributeN และชนิดของคุณลักษณะ ได้แก่ TypeAttr1, TypeAttr2, ..., TypeAttrN

กำหนดรูปแบบของคลาส แสดงได้ดังรูปที่ 3.4



รูปที่ 3.4 รูปแบบของคลาสในแผนภาพคลาส

ตัวอย่าง คลาส Customer เป็นคลาสที่กำหนดชื่อของคลาส ชื่อของคุณลักษณะ และชนิดของคุณลักษณะทั้งหมด แสดงดังรูปที่ 3.5



รูปที่ 3.5 ตัวอย่างคลาส Customer

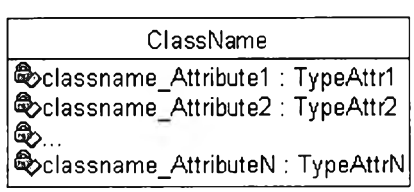
โดยในงานวิจัยได้ทำการนำเสนอกฎการสร้างแอ็บสแตร็คแมชชีนบีของคลาส ดังนี้

กฎข้อที่ 1

การสร้างแอ็บสแตร็คแมชชีนบีเบซิคคลาสที่แสดงคุณลักษณะและชนิดของคุณลักษณะทั้งหมดของคลาส

ในกฎข้อนี้จะทำการสร้างแอ็บสแตร็คแมชชีนบีเบซิคคลาส (BasicClass) ที่แสดงคุณลักษณะและชนิดของคุณลักษณะทั้งหมดของคลาส โดยทำการระบุชื่อของคลาสในอนุประโยค MACHINE ระบุเซตของคลาส ซึ่งระบุในอนุประโยค SETS ระบุตัวแปรคลาส และตัวแปรคุณลักษณะทั้งหมดในอนุประโยค VARIABLES ระบุเงื่อนไขของความสัมพันธ์ระหว่างตัวแปรคลาสกับเซตของคลาส และตัวแปรคุณลักษณะกับชนิดของคุณลักษณะทั้งหมดในอนุประโยค INVARIANT และระบุค่าเริ่มต้นของตัวแปรคลาสและตัวแปรคุณลักษณะทั้งหมดในอนุประโยค INITIALISATION

คลาสใด ๆ ในแผนภาพคลาส จะนำมาสร้างเป็นแอ็บสแตร็คแมชชีนบี โดยถือว่าใน 1 คลาสของแผนภาพคลาส ประกอบไปด้วยชื่อของคลาส คือ ClassName คุณลักษณะทั้งหมด คือ classname_Attribute1, classname_Attribute2, ..., classname_AttributeN และชนิดของคุณลักษณะทั้งหมด คือ TypeAttr1, TypeAttr2, ..., TypeAttrN จะแสดงในแอ็บสแตร็คแมชชีนบีเบซิคคลาส โดยถือว่าเป็น 1 แอ็บสแตร็คแมชชีนบีของเบซิคคลาส จะแสดงถึงคุณลักษณะและชนิดของคุณลักษณะทั้งหมดของคลาส สามารถพิจารณารูปแบบของคลาสที่แสดงคุณลักษณะและชนิดของคุณลักษณะทั้งหมด ดังรูปที่ 3.6



รูปที่ 3.6 รูปแบบคลาสที่ประกอบด้วยชื่อคลาส คุณลักษณะ และชนิดของคุณลักษณะทั้งหมด

จากรูปที่ 3.6 แสดงแอ็บสแตร็คแมชชีนบีเบซิคคลาส คือ BasicClassName ได้ดังรูปที่ 3.7

MACHINE
BasicClassName
...
SETS
BASICCLASSNAME
VARIABLES
basicclassname,
classname_Attribute1,
classname_Attribute2,
...,
classname_AttributeN
INVARIANT
basicclassname $\subset$ BASICCLASSNAME $\wedge$
classname_Attribute1 $\in$ TypeAttr1 $\wedge$
classname_Attribute2 $\in$ TypeAttr2 $\wedge$
... $\wedge$
classname_AttributeN $\in$ TypeAttrN
INITIALISATION
basicclassname := ϕ ||
classname_Attribute1 := InitialValue ||
classname_Attribute2 := InitialValue ||
...||
classname_AttributeN := InitialValue
END

รูปที่ 3.7 แอ็บสแตร็คแมชชีนเบซิคคลาส

โดยที่ BasicClassName

คือ ชื่อของแอ็บสแตร็คแมชชีนบีเบซิคคลาส

BASICCLASSNAME

คือ เซตของคลาสที่ระบุในแอ็บสแตร็คแมชชีนบีเบซิคคลาส โดยทำการระบุเซตของคลาสในอนุประโยค SETS

basicclassname

คือ ตัวแปรคลาสที่ระบุในแอ็บสแตร็คแมชชีนบีเบซิคคลาส โดยทำการระบุตัวแปรคลาสในอนุประโยค VARIABLES

classname_Attribute1, classname_Attribute2, classname_Attribute3, ... , classname_AttributeN

คือ ชื่อของตัวแปรคุณลักษณะทั้งหมดของคลาส โดยทำการระบุตัวแปรคุณลักษณะทั้งหมดในอนุประโยค VARIABLES

InitialValue

คือ ค่าเริ่มต้นของตัวแปรคุณลักษณะทั้งหมด โดยทำการระบุค่าเริ่มต้นของตัวแปรคุณลักษณะในอนุประโยค INITIALISATIONS

สามารถพิจารณาขั้นตอนวิธี (Algorithm) ของกฎข้อที่ 1 ได้ดังนี้

วิธีของกฎข้อที่ 1

สำหรับ Classi ใด ๆ ในแผนภาพคลาส โดยที่ Classi จะมีคุณลักษณะทั้งหมด ได้แก่ classi_Attribute1, classi_Attribute2, ..., classi_AttributeN ซึ่งแต่ละคุณลักษณะจะมีชนิดของคุณลักษณะ ได้แก่ TypeAttr1, TypeAttr2, ..., TypeAttrN ตามลำดับ

จากข้อความข้างต้น จะสามารถนำมาสร้างแอ็บสแตร็คแมชชีนบี BasicClassi ได้ดังนี้

R1-1) กำหนดอนุประโยค MACHINE

R1-2) กำหนดชื่อของแอ็บสแตร็คแมชชีนบี คือ BasicClassi ในอนุประโยค MACHINE

R1-3) กำหนดอนุประโยค SETS

R1-4) กำหนดเซตของคลาส Classi ของแอ็บสแตร็คแมชชีนบี BasicClassi คือ BASICCLASSi ในอนุประโยค SETS

R1-5) กำหนดอนุประโยค VARIABLES

R1-6) กำหนดตัวแปรคลาสของแอ็บสแตร็คแมชชีนบี BasicClassi คือ basicclassi และตัวแปรคุณลักษณะทั้งหมด คือ classi_Attribute1, classi_Attribute2, ..., classi_AttributeN ในอนุประโยค VARIABLES

R1-7) กำหนดอนุประโยค INVARIANT

R1-8) กำหนดความสัมพันธ์ระหว่างตัวแปรคลาสกับเซตของ Classi ในแอ็บสแตร็คแมชชีนบี BasicClassi นั่นคือ basicclassi $\subset$ BASICCLASSi .....(Inv1)

และกำหนดความสัมพันธ์ระหว่างตัวแปรของคุณลักษณะและชนิดของคุณลักษณะทั้งหมด นั่นคือ classi_Attribute1 $\in$ TypeAttr1.....(Inv2)

classi_Attribute2 $\in$ TypeAttr2.....(Inv3)

...

classi_AttributeN $\in$ TypeAttrN.....(InvN+1)

นำความสัมพันธ์ทั้งหมด ได้แก่ (Inv1), (Inv2), ..., (InvN+1) มาเชื่อมด้วยตัวเชื่อมตรรกคณิตศาสตร์ คือ ตัวเชื่อม "และ" นั่นคือ $\wedge$ จะได้ความสัมพันธ์ทั้งหมด คือ (Inv1) $\wedge$ (Inv2) $\wedge$ ... $\wedge$ (InvN+1) ในอนุประโยค INVARIANT

R1-9) กำหนดอนุประโยค INITIALISATION

R1-10) กำหนดค่าเริ่มต้นของตัวแปรคลาส นั่นคือ basicclassi := ϕ .....(Init1)

และกำหนดค่าเริ่มต้นของตัวแปรคุณลักษณะทั้งหมด นั่นคือ

classi_Attribute1 := InitialValue .....(Init2)

classi_Attribute2 := InitialValue .....(Init3)

...

classi_AttributeN := InitialValue .....(InitN+1)

นำความสัมพันธ์ทั้งหมด ได้แก่ (1), (2), ...,(N+1) มาเชื่อมด้วยเครื่องหมายการกำหนดค่าแบบขนาน (Parallel Assignment) นั่นคือ || จะได้ค่าเริ่มต้นของตัวแปรคลาสและเริ่มต้นของตัวแปรคุณลักษณะทั้งหมด คือ (1) || (2) || ... || (N+1) ในอนุประโยค INITIALISATION

R1-11) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี BasicClassi

จากรูปที่ 3.5 แสดงตัวอย่างของคลาส Customer เป็นคลาสที่มีคุณลักษณะทั้งหมด ได้แก่ รหัสของลูกค้า (customer_ID) ชื่อของลูกค้า (customer_Name) อายุของลูกค้า (customer_Age) ที่อยู่ของลูกค้า (customer_Address) หมายเลขโทรศัพท์ของลูกค้า (customer_PhoneNumber) และสถานภาพของลูกค้า (customer_Status) โดยมีชนิดของคุณลักษณะต่าง ๆ ดังนี้ customer_ID customer_Name customer_ Address และ customer_PhoneNumber มีชนิดคุณลักษณะเป็นสายอักขระ (STRING) customer_Age มีชนิดคุณลักษณะเป็นจำนวนเต็มที่มีค่าตั้งแต่ศูนย์ (NAT) และ customer_Status มีชนิดคุณลักษณะเป็นบูลีน (BOOLEAN) สามารถสร้างแอ็บสแตร็คแมชชีนบี BasicCustomer โดยทำการระบุคุณลักษณะ และชนิดของคุณลักษณะทั้งหมดของคลาส Customer ได้ดังรูปที่ 3.8

MACHINE
BasicCustomer
...
SETS
BASICCUSTOMER
VARIABLES
basiccustomer,
customer_ID,
customer_Name,
customer_Age,
customer_Address,
customer_PhoneNumber,
customer_Status
INVARIANT
basiccustomer ⊂ BASICCUSTOMER ∧
customer_ID ∈ STRING ∧
customer_Name ∈ STRING ∧
customer_Age ∈ NAT ∧
customer_Address ∈ STRING ∧
customer_PhoneNumber ∈ STRING ∧
customer_Status ∈ BOOLEAN
INITIALISATION
basiccustomer := ϕ ||
customer_ID := null ||
customer_Name := null ||
customer_Age := 0 ||
customer_Address := null ||
customer_PhoneNumber := null ||
customer_Status := FALSE
END

รูปที่ 3.8 แอ็บสแตร็คแมชชีนบี BasicCustomer

กฎข้อที่ 2

การสร้างแอ็บสแตร็คแมชชีนบีคลาส

ในกฎข้อนี้จะทำการสร้างแอ็บสแตร็คแมชชีนบีคลาสที่สามารถอ้างอิงคุณลักษณะทั้งหมดและชนิดของคุณลักษณะทั้งหมดของแอ็บสแตร็คแมชชีนบีเบซิคคลาส โดยในแอ็บสแตร็คแมชชีนบีคลาส จะมีอนุประโยค USES ทำให้สามารถอ้างถึงคุณลักษณะ และชนิดของคุณลักษณะทั้งหมดของแอ็บสแตร็คแมชชีนบีเบซิคคลาส แสดงรูปแบบของแอ็บสแตร็คแมชชีนบีคลาส ดังรูปที่ 3.9

```
MACHINE
Class
USES
BasicClass
...
END
```

รูปที่ 3.9 แอ็บสแตร็คแมชชีนบีคลาส

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 2 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 2

สำหรับ Classi ใด ๆ ในแผนภาพคลาส จะสามารถนำมาสร้างแอ็บสแตร็คแมชชีนบี Classi ได้ดังนี้

R2-1) กำหนดอนุประโยค MACHINE

R2-2) กำหนดชื่อของแอ็บสแตร็คแมชชีนบี คือ Classi ในอนุประโยค MACHINE

R2-3) กำหนดอนุประโยค USES

R2-4) ระบุชื่อของแอ็บสแตร็คแมชชีนบี คือ BasicClassi ในอนุประโยค USES

R2-5) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี Classi

ตัวอย่างของแอ็บสแตร็คแมชชีนบี Customer ที่สามารถอ้างถึงคุณลักษณะ และชนิดของคุณลักษณะทั้งหมดของแอ็บสแตร็คแมชชีนบี BasicCustomer แสดงได้ดังรูปที่ 3.10

```
MACHINE
Customer
USES
BasicCustomer
END
```

รูปที่ 3.10 แอ็บสแตร็คแมชชีนบีคลาส Customer

กฎข้อที่ 3

การสร้างไลบรารีแอ็บสแตร็คแมชชีนบี

ในกฎข้อนี้จะทำการสร้างไลบรารีแอ็บสแตร็คแมชชีนบี โดยไลบรารีแอ็บสแตร็คแมชชีนบีเป็นแอ็บสแตร็คแมชชีนบีที่แสดงเซตของชนิดคุณลักษณะ ประกอบด้วยไลบรารีแอ็บสแตร็คแมชชีนบี

สายอักขระ (StringType) แสดงชนิดของคุณลักษณะที่เป็นสายอักขระ และไลบรารีแอ็บสแตร็คแมชชีนบีบูลีน (BooleanType) แสดงชนิดของคุณลักษณะที่เป็นบูลีน โดยในแอ็บสแตร็คแมชชีนบีเบซิคคลาสและแอ็บสแตร็คแมชชีนบีคลาส จะมีอนุประโยค SEES ที่สามารถเห็นชนิดคุณลักษณะจากไลบรารีแอ็บสแตร็คแมชชีนบีสายอักขระและไลบรารีแอ็บสแตร็คแมชชีนบีบูลีน

แสดงรูปแบบของไลบรารีแอ็บสแตร็คแมชชีนบีทั้ง 2 ดังรูปที่ 3.11 และรูปที่ 3.12

```
MACHINE
StringType
SETS
STRING
VARIABLES
null,
EmptyString
INVARIANT
null ∈ STRING ∧
EmptyString ∈ STRING
INITIALISATION
null := EmptyString
END
```

รูปที่ 3.11 ไลบรารีแอ็บสแตร็คแมชชีนบีสายอักขระ

```
MACHINE
BooleanType
SETS
BOOLEAN = {TRUE, FALSE}
END
```

รูปที่ 3.12 ไลบรารีแอ็บสแตร็คแมชชีนบีบูลีน

สำหรับคุณลักษณะที่มีชนิดคุณลักษณะเป็นสายอักขระ จะทำการระบุตัวแปรคุณลักษณะเหล่านี้ในอนุประโยค VARIABLES ของแอ็บสแตร็คแมชชีนบีเบซิคคลาส เพื่ออธิบายคุณลักษณะที่มีชนิดคุณลักษณะเป็นสายอักขระ ทำการระบุความสัมพันธ์ของตัวแปรคุณลักษณะกับชนิดตัวแปรคุณลักษณะที่เป็นสายอักขระในอนุประโยค INVARIANT และระบุค่าเริ่มต้นของตัวแปรคุณลักษณะเหล่านี้ โดยทำการระบุค่าเริ่มต้นเป็นนัล (null) ในอนุประโยค INITIALISATION โดยแอ็บสแตร็คแมชชีนเบซิคคลาสจะมีอนุประโยค SEES ไลบรารีแอ็บสแตร็คแมชชีนบีสายอักขระ ทำให้มองเห็นรูปแบบชนิดของคุณลักษณะที่เป็นสายอักขระจากไลบรารีแอ็บสแตร็คแมชชีนบีสายอักขระแสดงได้ดังรูปที่ 3.13

สำหรับชนิดคุณลักษณะที่เป็นบูลีน จะทำการระบุตัวแปรคุณลักษณะเหล่านี้ในอนุประโยค VARIABLES ของแอ็บสแตร็คแมชชีนบีเบซิคคลาส เพื่ออธิบายถึงคุณลักษณะที่มีชนิดของคุณลักษณะเป็นบูลีน ทำการระบุความสัมพันธ์ของตัวแปรคุณลักษณะกับชนิดของคุณลักษณะที่เป็นบูลีนในอนุประโยค INVARIANT และระบุค่าเริ่มต้นของตัวแปรคุณลักษณะที่เป็นบูลีน โดยทำการระบุค่า

เริ่มต้นเป็นเท็จ (FALSE) ในอนุประโยค INITIALISATION โดยแอ็บสแตร็คแมชชีนบีเบซิคคลาสจะมีอนุประโยค SEES ไลบรารีแอ็บสแตร็คแมชชีนบีบูลีน ทำให้มองเห็นรูปแบบชนิดของคุณลักษณะที่เป็นบูลีนจากไลบรารีแอ็บสแตร็คแมชชีนบีบูลีน แสดงได้ดังรูปที่ 3.14

```
MACHINE
BasicClass
SEES
StringType
VARIABLES
classname_Attribute1,
classname_Attribute2,
...,
classname_AttributeN
INVARIANT
classname_Attribute1 ∈ STRING ∧
classname_Attribute2 ∈ STRING ∧
... ∧
classname_AttributeN ∈ STRING
INITIALISATION
classname_Attribute1 := null ||
classname_Attribute2 := null ||
...||
classname_AttributeN:= null
...
END
```

รูปที่ 3.13 แอ็บสแตร็คแมชชีนบีเบซิคคลาสที่มีชนิดของคุณลักษณะเป็นสายอักขระ

```
MACHINE
BasicClass
SEES
BooleanType
VARIABLES
classname_Attribute1,
classname_Attribute2,
...,
classname_AttributeN
INVARIANT
classname_Attribute1 ∈ BOOLEAN ∧
classname_Attribute2 ∈ BOOLEAN ∧
... ∧
classname_AttributeN ∈ BOOLEAN
INITIALISATION
classname_Attribute1 := FALSE ||
classname_Attribute2 := FALSE ||
...||
classname_AttributeN:= FALSE
...
END
```

รูปที่ 3.14 แอ็บสแตร็คแมชชีนบีเบซิคคลาสที่มีชนิดของคุณลักษณะเป็นบูลีน

อาศัยกฎการสร้างแอ็บสแตร็คแมชชีนบีเบซิคคลาส กฎการสร้างแอ็บสแตร็คแมชชีนบีคลาส และกฎการสร้างไลบรารีแอ็บสแตร็คแมชชีนบี จะได้แอ็บสแตร็คแมชชีนบีทั้งหมด ดังนี้

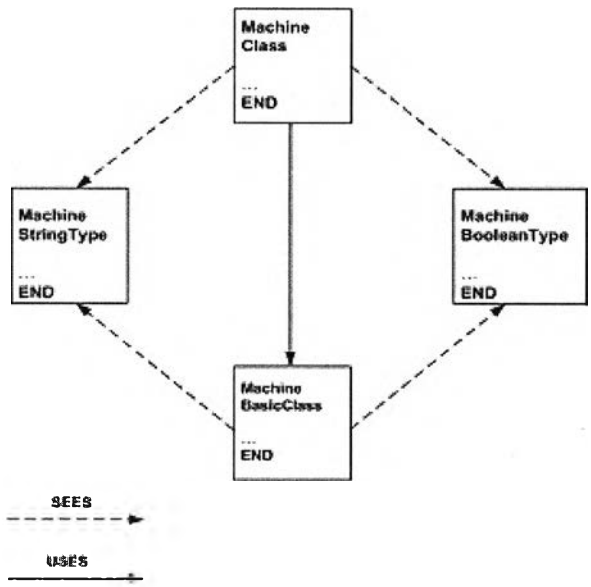
1) แอ็บสแตร็คแมชชีนบีคลาส

2) แอ็บสแตร็คแมชชีนบีเบซิคคลาส

3) ไลบรารีแอ็บสแตร็คแมชชีนบีสายอักขระ

4) ไลบรารีแอ็บสแตร็คแมชชีนบีบูลีน

สามารถแสดงรูปแบบของสถาปัตยกรรมแอ็บสแตร็คแมชชีนบีทั้งหมด ได้ดังรูปที่ 3.15



รูปที่ 3.15 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบี และไลบรารีแอ็บสแตร็คแมชชีนบี

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 3 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 3

สำหรับ Classi ใด ๆ ในแผนภาพคลาส โดยที่ Classi จะมีคุณลักษณะทั้งหมด ได้แก่ classi_Attribute1, classi_Attribute2, ..., classi_AttributeN ซึ่งแต่ละคุณลักษณะจะมีชนิดของคุณลักษณะ ได้แก่ TypeAttr1, TypeAttr2, ..., TypeAttrN ตามลำดับ

จากข้อความข้างต้น จะสามารถนำมาสร้างไลบรารีแอ็บสแตร็คแมชชีนบี คือ StringType และ BooleanType โดยแอ็บสแตร็คแมชชีนบี BasicClassi จะสามารถระบุไลบรารีแอ็บสแตร็คแมชชีนบีในอนุประโยค SEES ได้ดังนี้

R3-1) กำหนดอนุประโยค MACHINE

R3-2) กำหนดชื่อของแอ็บสแตร็คแมชชีนบี คือ BasicClassi ในอนุประโยค MACHINE

R3-3) กำหนดอนุประโยค SEES

R3-4) ถ้าชนิดของคุณลักษณะ TypeAttr1 หรือ TypeAttr2 หรือ ... หรือ TypeAttrN เป็น STRING แล้วทำการระบุ StringType ในอนุประโยค SEES

R3-5) ถ้าชนิดของคุณลักษณะ TypeAttr1 หรือ TypeAttr2 หรือ ... หรือ TypeAttrN เป็น BOOLEAN

แล้วทำการระบุ BooleanType ในอนุประโยค SEES

R3-6) กำหนดอนุประโยค INVARIANT

R3-7) จากข้อที่ R3-4 จะสามารถทำการสร้างความสัมพันธ์ระหว่างตัวแปรคุณลักษณะกับชนิดของคุณลักษณะทั้งหมดที่เป็น STRING นั่นคือ

นั่นคือ

classi_Attribute1 ∈ STRING ∧

classi_Attribute2 ∈ STRING ∧

... ∧

classi_AttributeN ∈ STRING ในอนุประโยค INVARIANT

จากข้อที่ R3-5 จะสามารถทำการสร้างความสัมพันธ์ระหว่างตัวแปรคุณลักษณะกับชนิดของคุณลักษณะทั้งหมดที่เป็น BOOLEAN นั่นคือ

นั่นคือ

classi_Attribute1 ∈ BOOLEAN ∧

classi_Attribute2 ∈ BOOLEAN ∧

... ∧

classi_AttributeN ∈ BOOLEAN ในอนุประโยค INVARIANT

R3-8) กำหนดอนุประโยค INITIALISATION

R3-9) จากข้อที่ R3-4 จะสามารถทำการระบุค่าเริ่มต้นของตัวแปรคุณลักษณะทั้งหมดที่เป็น STRING นั่นคือ

classi_Attribute1 := null ||

classi_Attribute2 := null ||

... ||

classi_AttributeN := null ในอนุประโยค INITIALISATION

จากข้อที่ R3-5 จะสามารถทำการระบุค่าเริ่มต้นของตัวแปรคุณลักษณะทั้งหมดที่เป็น BOOLEAN นั่นคือ

classi_Attribute1 := FALSE ||

classi_Attribute2 := FALSE ||

... ||

classi_AttributeN := FALSE ในอนุประโยค INITIALISATION

R3-10) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี BasicClassi

จากตัวอย่างแอ็บสแตร็คแมชชีนบี BasicCustomer จากรูปที่ 3.8 ในอนุประโยค SEES จะระบุไลบรารีแอ็บสแตร็คแมชชีนบี ทำให้เห็นรูปแบบชนิดของคุณลักษณะที่เป็นสายอักขระและบูลีน แสดงได้ดังรูปที่ 3.16

```
MACHINE
BasicCustomer
SEES
StringType,
BooleanType
...
END
```

รูปที่ 3.16 แอ็บสแตร็คแมชชีนบี BasicCustomer ที่มีอนุประโยค SEES ไลบรารีแอ็บสแตร็คแมชชีนบี

3.2.2 การสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชัน ความสัมพันธ์แอกริเกชัน และความสัมพันธ์คอมโพสิชันของคลาส

การสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชัน ความสัมพันธ์แอกริเกชัน และความสัมพันธ์คอมโพสิชันของคลาส โดยการระบุมัลติพลิซิตีของแต่ละคลาสจากความสัมพันธ์ทั้ง 3 ทำการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์ โดยมีคำมาเติมข้างหน้า (Prefix) แล้วตามด้วยชื่อของความสัมพันธ์ในแต่ละชนิด ถ้าเป็นความสัมพันธ์แอสโซซิเอชันจะมีคำเติมข้างหน้า คือ Asso ถ้าเป็นความสัมพันธ์แอกริเกชันจะมีคำเติมข้างหน้า คือ Aggr และความสัมพันธ์คอมโพสิชันจะมีคำเติมข้างหน้า คือ Compo

ในงานวิจัยนี้จะกล่าวถึงมัลติพลิซิตีของคลาสที่มีความสัมพันธ์ระหว่างกัน โดยนำเสนอแนวคิดเกี่ยวกับมัลติพลิซิตี ดังนี้

มัลติพลิซิตี (Multiplicity)

รูปแบบของมัลติพลิซิตีจะสามารถเขียนอยู่ในรูปแบบของช่วงหรือค่าที่แน่นอน ซึ่งเป็นการระบุจำนวนออบเจ็คของคลาสที่มีความสัมพันธ์ระหว่างกัน โดยที่ Asso_Class1_Class2 หรือ Aggr_Class1_Class2 หรือ Compo_Class1_Class2 เป็นความสัมพันธ์แอสโซซิเอชัน หรือ ความสัมพันธ์แอกริเกชัน หรือ ความสัมพันธ์คอมโพสิชัน ระหว่าง Class1 และ Class2 ตามลำดับ

โดยที่ * คือ More

0...* คือ Zero or More

1...* คือ One or More

1 คือ One

0...1 คือ Zero or One

ค่าของมัลติพลิซิตี * หรือ 0...* หรือ 1...* หรือ 1 หรือ 0...1 สามารถระบุในรูปแบบเพรดิเคต เพื่อระบุเงื่อนไขของความสัมพันธ์เพิ่มเติมในส่วนของอนุประโยค INVARIANT ได้ดังนี้

กำหนดให้ RelTypeRelName_class1_class2

คือ ตัวแปรในแอ็บสแตร็คแมชชีนบีที่แสดงถึงชนิดและชื่อของความสัมพันธ์ระหว่าง Class1 และ Class2

โดยที่ RelTypeRelName_class1_class2 $\subset$ BASICCLASS1 $\times$ BASICCLASS2

คือ ตัวแปรในแอ็บสแตร็คแมชชีนบีที่แสดงถึงชนิดและชื่อความสัมพันธ์ระหว่าง Class1 และ Class2 ซึ่งเป็นสับเซตของผลคูณคาร์ทีเซียนของ BASICCLASS1 และ BASICCLASS2

พิจารณาถึงการระบุค่าของมัลติพลิซิตี ดังนี้

- **ค่าของมัลติพลิซิตี** *

เงื่อนไขของมัลติพลิซิตีของ Class1 และ Class2 ในความสัมพันธ์ระหว่าง Class1 กับ Class2 ให้เขียนเป็น

$$\begin{aligned}
&\text{dom(RelTypeRelName_class1_class2)} = \text{basicclass1} \wedge \\
&\text{ran(RelTypeRelName_class1_class2)} = \text{basicclass2} \wedge \\
&\forall (xx,yy).(((xx \in \text{dom(RelTypeRelName_class1_class2)}) \wedge \\
&\qquad (yy \in \text{ran(RelTypeRelName_class1_class2)})) \\
&\qquad \rightarrow \text{card}((\text{RelTypeRelName_class1_class2})[\{xx\}]) \geq 0 \wedge \\
&\qquad \text{card}((\text{RelTypeRelName_class1_class2})^{-1}[\{yy\}]) \geq 0)
\end{aligned}$$

- **ค่าของมัลติพลิซิตี** 0...*

เงื่อนไขของมัลติพลิซิตีของ Class1 และ Class2 ในความสัมพันธ์ระหว่าง Class1 กับ Class2 ให้เขียนเป็น

$$\begin{aligned}
&\text{dom(RelTypeRelName_class1_class2)} = \text{basicclass1} \wedge \\
&\text{ran(RelTypeRelName_class1_class2)} = \text{basicclass2} \wedge \\
&\forall (xx,yy).(((xx \in \text{dom(RelTypeRelName_class1_class2)}) \wedge \\
&\qquad (yy \in \text{ran(RelTypeRelName_class1_class2)})) \\
&\qquad \rightarrow \text{card}((\text{RelTypeRelName_class1_class2})[\{xx\}]) \geq 0 \wedge \\
&\qquad \text{card}((\text{RelTypeRelName_class1_class2})^{-1}[\{yy\}]) \geq 0)
\end{aligned}$$

- **ค่าของมัลติพลิซิตี** 1...*

เงื่อนไขของมัลติพลิซิตีของ Class1 และ Class2 ในความสัมพันธ์ระหว่าง Class1 กับ Class2 ให้เขียนเป็น

$$\text{dom(RelTypeRelName_class1_class2)} = \text{basicclass1} \wedge$$

ran(RelTypeRelName_class1_class2) = basicclass2 ∧

∀ (xx,yy).(((xx ∈ dom(RelTypeRelName_class1_class2)) ∧

(yy ∈ ran(RelTypeRelName_class1_class2)))

→ card((RelTypeRelName_class1_class2)[{xx}]) ≥ 1 ∧

card((RelTypeRelName_class1_class2)$^{-1}$[{yy}]) ≥ 1)

- **ค่าของมัลติพลิซิตี** 1

เงื่อนไขของมัลติพลิซิตีของ Class1 และ Class2 ในความสัมพันธ์ระหว่าง Class1 กับ Class2 ให้เขียนเป็น

dom(RelTypeRelName_class1_class2) = basicclass1 ∧

ran(RelTypeRelName_class1_class2) = basicclass2 ∧

∀ (xx,yy).(((xx ∈ dom(RelTypeRelName_class1_class2)) ∧

(yy ∈ ran(RelTypeRelName_class1_class2)))

→ card((RelTypeRelName_class1_class2)[{xx}]) = 1 ∧

card((RelTypeRelName_class1_class2)$^{-1}$[{yy}]) = 1)

- **ค่าของมัลติพลิซิตี** 0...1

เงื่อนไขของมัลติพลิซิตีของ Class1 และ Class2 ในความสัมพันธ์ระหว่าง Class1 กับ Class2 ให้เขียนเป็น

dom(RelTypeRelName_class1_class2) = basicclass1 ∧

ran(RelTypeRelName_class1_class2) = basicclass2 ∧

∀ (xx,yy).(((xx ∈ dom(RelTypeRelName_class1_class2)) ∧

(yy ∈ ran(RelTypeRelName_class1_class2)))

→ card((RelTypeRelName_class1_class2)[{xx}]) ≥ 0 ∧

card((RelTypeRelName_class1_class2)[{xx}]) ≤ 1 ∧

card((RelTypeRelName_class1_class2)$^{-1}$[{yy}]) ≥ 0 ∧

card((RelTypeRelName_class1_class2)$^{-1}$[{yy}]) ≤ 1)

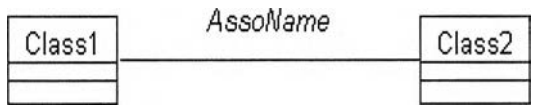
งานวิจัยนี้ได้ออกแบบกฎการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชัน ความสัมพันธ์แอกริเกชัน และความสัมพันธ์คอมโพสิชัน ดังนี้

กฎข้อที่ 4

การสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันแบบทวิภาคระหว่างคลาส

ในกฎข้อนี้จะทำการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันแบบทวิภาคระหว่างคลาส โดยทุก ๆ ความสัมพันธ์แอสโซซิเอชันแบบทวิภาคจะนำมาสร้างเป็นแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาส นั่นคือ

จากความสัมพันธ์แอสโซซิเอชันระหว่างคลาส 2 คลาสในแผนภาพคลาส ซึ่ง Class1 และ Class2 ที่มีความสัมพันธ์แบบแอสโซซิเอชัน โดยมีชื่อของความสัมพันธ์คือ AssoName แสดงได้ดังรูปที่ 3.17



รูปที่ 3.17 รูปแบบความสัมพันธ์แบบแอสโซซิเอชัน "AssoName" ระหว่าง Class1 กับ Class2

จากรูปที่ 3.17 สามารถสร้างแอ็บสแตร็คแมชชีนบี BasicClass1 แอ็บสแตร็คแมชชีนบี Class1 แอ็บสแตร็คแมชชีนบี BasicClass2 และแอ็บสแตร็คแมชชีนบี Class2 โดยที่แอ็บสแตร็คแมชชีน Class1 และแอ็บสแตร็คแมชชีน Class2 จะระบุถึงแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่าง Class1 และ Class2 ในอนุประโยค USES จะแสดงได้ดังรูปที่ 3.18 และ รูปที่ 3.19

```
MACHINE
Class1
USES
BasicClass1,
AssoAssoName_Class1_Class2
...
END
```

รูปที่ 3.18 แอ็บสแตร็คแมชชีน Class1 ระบุแอ็บสแตร็คแมชชีนบี AssoAssoName_Class1_Class2 ในอนุประโยค USES

MACHINE
Class2
USES
BasicClass2,
AssoAssoName_Class1_Class2
...
END

รูปที่ 3.19 แอ็บสแตร็คแมชชีน Class2 ระบุแอ็บสแตร็คแมชชีนบี AssoAssoName_Class1_Class2 ในอนุประโยค USES

ทำการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสทั้ง 2 แสดงได้ดังรูปที่ 3.20

MACHINE
AssoAssoName_Class1_Class2
USES
BasicClass1, BasicClass2
VARIABLES
assoassoname_class1_class2
INVARIANT
assoassoname_class1_class2 $\subset$ BASICCLASS1 $\times$ BASICCLASS2 $\wedge$
dom(assoassoname_class1_class2) = basicclass1 $\wedge$
ran(assoassoname_class1_class2) = basicclass2 $\wedge$
$\forall$ (xx,yy).(((xx $\in$ dom(assoassoname_class1_class2)) $\wedge$
(yy $\in$ ran(assoassoname_class1_class2)))
$\rightarrow$ card((assoassoname_class1_class2)[{xx}]) = Cardinality of Class2 $\wedge$
card((assoassoname_class1_class2)$^{-1}$[{yy}]) = Cardinality of Class1)
INITIALISATION
assoassoname_class1_class2 := ϕ
END

รูปที่ 3.20 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสทั้ง 2 คลาส

โดยที่ assoassoname_class1_class2

คือ ตัวแปรในแอ็บสแตร็คแมชชีนบีที่แสดงถึงความสัมพันธ์แอสโซซิเอชันโดยมีคำนำหน้า asso ติดกับชื่อของความสัมพันธ์คือ AssoName แล้วตามด้วยชื่อของ Class1 และ Class2 โดยชื่อของความสัมพันธ์แอสโซซิเอชันและคลาสทั้ง 2 ต้องใช้อักษรภาษาอังกฤษตัวพิมพ์เล็ก

dom และ ran

คือ การระบุโดเมน และพิสัยของความสัมพันธ์แอสโซซิเอชัน

xx และ yy

คือ ตัวแปรที่เป็นสมาชิกของโดเมน และพิสัยของความสัมพันธ์แอสโซซิเอชัน

card((assoassoname_class1_class2) [{xx}])

คือ จำนวนสมาชิกของโดเมนจากความสัมพันธ์แอสโซซิเอชันระหว่าง Class1 และ Class2

card((assoassoname_class1_class2)$^{-1}$ [{yy}])

คือ จำนวนสมาชิกของส่วนผกผันพิสัยจากความสัมพันธ์แอสโซซิเอชันระหว่าง Class1 และ Class2

Cardinality of Class1

คือ จำนวนออบเจ็คของ Class1

Cardinality of Class2

คือ จำนวนออบเจ็คของ Class2

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 4 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 4

สำหรับ Classi และ Classj ใด ๆ ในแผนภาพคลาส โดยมีความสัมพันธ์แอสโซซิเอชัน AssoName ระหว่าง Classi และ Classj

จากข้อความข้างต้น จะทำการสร้างแอ็บสแตร็คแมชชีนบี AssoAssoName_Classi_Classj ได้ดังนี้

R4-1) กำหนดอนุประโยค MACHINE

R4-2) กำหนดชื่อของแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่าง Classi และ Classj คือ AssoAssoName_Classi_Classj ในอนุประโยค MACHINE

R4-3) กำหนดอนุประโยค USES

R4-4) ระบุชื่อของแอ็บสแตร็คแมชชีนบี BasicClassi และ BasicClassj ในอนุประโยค USES

R4-5) จากข้อที่ R4-1 กำหนดเซตของตัวแปรของความสัมพันธ์แอสโซซิเอชันระหว่าง Classi และ Classj คือ assoassoname_classi_classj ในอนุประโยค VARIABLES

R4-6) กำหนดความสัมพันธ์ระหว่างเซตของตัวแปรของความสัมพันธ์แอสโซซิเอชันระหว่าง Classi และ Classj คือ assoassoname_classi_classj และผลคูณคาร์ทีเซียนระหว่างเซตตัวแปรของแอ็บสแตร็คแมชชีน BasicClassi และ BasicClassj นั่นคือ BASICCLASSi × BASICCLASSj จะได้ว่า assoassoname_classi_classj ⊂ BASICCLASSi × BASICCLASSj .....(1)

กำหนด dom(assoassoname_classi_classj) = basicclassi.....(2)

และ ran(assoassoname_classi_classj) = basicclassj.....(3)

กำหนดตัวแปร xx และ yy ภายใต้ตัวบ่งปริมาณ"ทั้งหมด" (For All) นั่นคือ ∀ (xx,yy)

ทำการสร้างเงื่อนไขของตัวแปร xx และ yy คือ

xx ∈ dom(assoassoname_classi_classj) ∧ yy ∈ ran(assoassoname_classi_classj)

ภายใต้ตัวบ่งปริมาณทั้งหมดของตัวแปร xx และ yy นั่นคือ

$\forall$ (xx,yy) (xx $\in$ dom(assoassoname_classi_classj) $\wedge$

yy $\in$ ran(assoassoname_classi_classj))

ทำการกำหนดจำนวนเซตของตัวแปรของความสัมพันธ์แอสโซซิเอชันระหว่าง Classi และ Classj หรือ assoassoname_classi_classj ของตัวแปร xx และ yy เพื่อระบุมัลติพลิซิตีของ Classi และ Classj นั่นคือ

card((assoassoname_classi_classj)[{xx}]) แทนมัลติพลิซิตีของ Classj และ

card((assoassoname_classi_classj)$^{-1}$[{yy}]) แทนมัลติพลิซิตีของ Classi

โดยมัลติพลิซิตีของ Classi และ Classj จะเป็นผลจากการระบุเงื่อนไขของตัวแปร xx และ yy ภายใต้ภายใต้ตัวบ่งปริมาณทั้งหมดของตัวแปร xx และ yy นั่นคือ

$\forall$ (xx,yy) (((xx $\in$ dom(assoassoname_classi_classj) $\wedge$

(yy $\in$ ran(assoassoname_classi_classj))

$\rightarrow$ card((assoassoname_classi_classj)[{xx}]) = Multiplicity of Classj $\wedge$

card((assoassoname_classi_classj)$^{-1}$[{yy}]) = Multiplicity of Classi)

.....(4)

นำ (1) $\wedge$ (2) $\wedge$ (3) $\wedge$ (4) จะได้ว่า

assoassoname_classi_classj $\subset$ BASICCLASSi $\times$ BASICCLASSj $\wedge$

dom(assoassoname_classi_classj) = basicclassi $\wedge$

ran(assoassoname_classi_classj) = basicclassj $\wedge$

$\forall$ (xx,yy) (((xx $\in$ dom(assoassoname_classi_classj) $\wedge$

(yy $\in$ ran(assoassoname_classi_classj))

$\rightarrow$ card((assoassoname_classi_classj)[{xx}]) = Multiplicity of Classj $\wedge$

card((assoassoname_classi_classj)$^{-1}$[{yy}]) = Multiplicity of Classi)

ในอนุประโยค INVARIANT

R4-7) กำหนดอนุประโยค INITIALISATION

R4-8) กำหนดค่าเริ่มต้นของเซตของตัวแปรของความสัมพันธ์แอสโซซิเอชันระหว่าง Classi และ Classj คือ assoassoname_classi_classj := ϕ ในอนุประโยค INITIALISATION

R4-9) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี AssoAssoName_Classi_Classj

R4-10) ระบุแอ็บสแตร็คแมชชีนบี AssoAssoName_Classi_Classj ในอนุประโยค USES ของแอ็บสแตร็คแมชชีนบี Classi และ Classj

ตัวอย่างการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชัน คือ คลาส Customer จะมีความสัมพันธ์แอสโซซิเอชันกับคลาส Order นั่นคือคลาส Customer แต่ละคลาสสามารถมีความสัมพันธ์แอสโซซิเอชันกับคลาส Order ได้ตั้งแต่หนึ่งขึ้นไป หรือ 1...* ในทำนองเดียวกันกับคลาส Order แต่ละคลาสสามารถมีความสัมพันธ์แอสโซซิเอชันกับคลาส Customer ได้เพียงหนึ่งเท่านั้น หรือ 1 แสดงได้ดังรูปที่ 3.21

รูปที่ 3.21 ตัวอย่างความสัมพันธ์แอสโซซิเอชันระหว่างคลาส Customer กับคลาส Order

จากรูปที่ 3.21 สามารถสร้างแอ็บสแตร็คแมชชีนบี BasicCustomer แอ็บสแตร็คแมชชีนบี Customer แอ็บสแตร็คแมชชีนบี BasicOrder และแอ็บสแตร็คแมชชีนบี Order โดยที่แอ็บสแตร็คแมชชีน Customer และแอ็บสแตร็คแมชชีน Order จะระบุถึงแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่าง Customer และ Order ในอนุประโยค USES จะแสดงได้ดังรูปที่ 3.22 และ รูปที่ 3.23

```
MACHINE
Customer
USES
BasicCustomer,
Asso_Customer_Order
...
END
```

รูปที่ 3.22 แอ็บสแตร็คแมชชีน Customer ระบุถึงแอ็บสแตร็คแมชชีนบี Asso_Customer_Order ในอนุประโยค USES

```
MACHINE
Order
USES
BasicOrder,
Asso_Customer_Order
...
END
```

รูปที่ 3.23 แอ็บสแตร็คแมชชีน Order ระบุถึงแอ็บสแตร็คแมชชีนบี Asso_Customer_Order ในอนุประโยค USES

สามารถสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาส Customer กับคลาส Order แสดงได้ดังรูปที่ 3.24

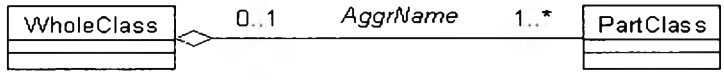
MACHINE
Asso_Customer_Order
USES
BasicCustomer, BasicOrder
VARIABLES
asso_customer_order
INVARIANT
asso_customer_order $\subset$ BASICCUSTOMER $\times$ BASICORDER $\wedge$
dom(asso_customer_order) = basiccustomer $\wedge$
ran(asso_customer_order) = basicorder $\wedge$
$\forall$ (xx,yy).(((xx $\in$ dom(asso_customer_order)) $\wedge$
(yy $\in$ ran(asso_customer_order)))
$\rightarrow$ card((asso_customer_order)[{xx}]) $\geq$ 1 $\wedge$
card((asso_customer_order)$^{-1}$[{yy}]) = 1)
INITIALISATION
asso_customer_order := ϕ
END

รูปที่ 3.24 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาส Customer กับคลาส Order

กฎข้อที่ 5
การสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาสหลักกับคลาสย่อย

ในกฎข้อนี้จะทำการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาสหลักกับคลาสย่อย โดยทุก ๆ ความสัมพันธ์แอกริเกชันจะนำมาสร้างเป็นแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาสหลักกับคลาสย่อย นั่นคือ

จากความสัมพันธ์แอกริเกชันระหว่างคลาสหลักกับคลาสย่อยในแผนภาพคลาส ซึ่งคลาสหลักและคลาสย่อยที่มีความสัมพันธ์แอกริเกชัน มีชื่อของความสัมพันธ์คือ AggrName แสดงได้ดังรูปที่ 3.25



รูปที่ 3.25 รูปแบบความสัมพันธ์แอกริเกชันระหว่างคลาสหลักกับคลาสย่อย

จากรูปแบบของความสัมพันธ์แอกริเกชันระหว่างคลาสหลักกับคลาสย่อย คลาสหลักจะทำการระบุรหัสอ้างถึง (Reference ID) คลาสย่อยแต่ละคลาส นั่นคือ คลาสหลักจะเก็บรหัสของคลาสย่อย เพื่อให้คลาสหลักสามารถอ้างถึงคลาสย่อยได้ ซึ่งการระบุรหัสอ้างถึงนั้น จะทำการระบุตัวแปรรหัสอ้างถึงคลาสย่อยในอนุประโยค VARIABLES สร้างความสัมพันธ์ที่มีเงื่อนไขที่เป็นจริงของตัวแปรรหัสอ้างถึงคลาสย่อยในอนุประโยค INVARIANT และทำการระบุค่าเริ่มต้นตัวแปรรหัสอ้างถึงคลาสย่อยในอนุประโยค INITIALISATION ในแอ็บสแตร็คแมชชีนบีเบซิคคลาสหลัก

แสดงแอ็บสแตร็คแมชชีนบี WholeClass แอ็บสแตร็คแมชชีนบี BasicWholeClass แอ็บสแตร็คแมชชีนบี PartClass และแอ็บสแตร็คแมชชีนบี BasicPartClass

โดยที่แอ็บสแตร็คแมชชีนบี WholeClass และแอ็บสแตร็คแมชชีนบี PartClass จะระบุถึงแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาสหลักและคลาสย่อย คือ AggrAggrName_WholeClass_PartClass ในอนุประโยค USES แสดงได้ดังรูปที่ 3.26 ถึงรูปที่ 3.29

```
MACHINE
WholeClass
USES
BasicWholeClass,
AggrAggrName_WholeClass_PartClass
...
END
```

รูปที่ 3.26 แอ็บสแตร็คแมชชีนบี WholeClass

```
MACHINE
BasicWholeClass
SEES
StringType
VARIABLES
...,
RefPartClassID
INVARIANT
... ∧
RefPartClassID ∈ STRING
INITIALISATION
...||
RefPartClassID := null
END
```

รูปที่ 3.27 แอ็บสแตร็คแมชชีนบี BasicWholeClass

```
MACHINE
PartClass
USES
BasicPartClass,
AggrAggrName_WholeClass_PartClass
...
END
```

รูปที่ 3.28 แอ็บสแตร็คแมชชีนบี PartClass

```
MACHINE
BasicPartClass
...
END
```

รูปที่ 3.29 แอ็บสแตร็คแมชชีนบี BasicPartClass

โดยที่ RefPartClassID

คือ ตัวแปรรหัสอ้างถึงคลาสย่อยที่ระบุในแอ็บสแตร็คแมชชีนบี BasicWholeClass

สร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาสหลัก และคลาสย่อย แสดงได้ดังรูปที่ 3.30

MACHINE
AggrAggrName_WholeClass_PartClass
USES
BasicWholeClass, BasicPartClass
VARIABLES
aggraggrname_wholeclass_partclass
INVARIANT
aggraggrname_wholeclass_partclass $\subset$ BASICWHOLECLASS $\times$ BASICPARTCLASS $\wedge$
dom(aggraggrname_wholeclass_partclass) = basicwholeclass $\wedge$
ran(aggraggrname_wholeclass_partclass) = basicpartclass $\wedge$
$\forall$ (xx,yy).(((xx $\in$ dom(aggraggrname_wholeclass_partclass)) $\wedge$
(yy $\in$ ran(aggraggrname_wholeclass_partclass)))
$\rightarrow$ card((aggraggrname_wholeclass_partclass)[{xx}]) $\geq$ 1 $\wedge$
card((aggraggrname_wholeclass_partclass)$^{-1}$[{yy}]) $\geq$0 $\wedge$
card((aggraggrname_wholeclass_partclass)$^{-1}$[{yy}]) $\leq$ 1)
INITIALISATION
aggraggrname_wholeclass_partclass:= ϕ
END

รูปที่ 3.30 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสหลักกับคลาสย่อย

โดยที่ aggraggrname_wholeclass_partclass

คือ ตัวแปรในแอ็บสแตร็คแมชชีนบีที่แสดงถึงความสัมพันธ์แอกริเกชันโดยมีคำนำหน้า aggr ติดกับชื่อของความสัมพันธ์ คือ AggrName แล้วตามด้วยชื่อของ WholeClass และ PartClass โดยชื่อของความสัมพันธ์แอกริเกชันและคลาสทั้ง 2 ต้องใช้อักษรภาษาอังกฤษตัวพิมพ์เล็ก

dom และ ran

คือ การระบุโดเมน และพิสัยของความสัมพันธ์แอกริเกชัน

xx และ yy

คือ ตัวแปรที่เป็นสมาชิกของโดเมน และพิสัยของความสัมพันธ์แอกริเกชัน

card((aggraggrname_wholeclass_partclass) [{xx}])

คือ จำนวนสมาชิกของโดเมนของความสัมพันธ์แอกริเกชันระหว่าง WholeClass และ PartClass

card((aggraggrname_wholeclass_partclass)$^{-1}$[{yy}])

คือ จำนวนสมาชิกของส่วนผกผันพิสัยของความสัมพันธ์แอกริเกชันระหว่าง WholeClass และ PartClass

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 5 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 5

สำหรับ WholeClassi และ PartClassi ใด ๆ ในแผนภาพคลาส โดยมีความสัมพันธ์แอกริเกชัน AggrName ระหว่าง WholeClassi และ PartClassi

จากข้อความข้างต้น จะทำการสร้างแอ็บสแตร็คแมชชีนบี AggrAggrName_WholeClassi_PartClassi ได้ดังนี้

R5-1) กำหนดอนุประโยค MACHINE

R5-2) กำหนดชื่อของแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่าง WholeClassi และ PartClassi คือ AggrAggrName_WholeClassi_PartClassi ในอนุประโยค MACHINE

R5-3) กำหนดอนุประโยค USES

R5-4) ระบุชื่อของแอ็บสแตร็คแมชชีนบี BasicWholeClassi และ BasicPartClassi ในอนุประโยค USES

R5-5) จากข้อที่ R5-1 กำหนดเซตของตัวแปรของความสัมพันธ์แอกริเกชันระหว่าง WholeClassi และ PartClassi คือ aggraggrname_wholeclassi_partclassi ในอนุประโยค VARIABLES

R5-6) กำหนดความสัมพันธ์ระหว่างเซตของตัวแปรของความสัมพันธ์แอกริเกชันระหว่าง WholeClassi และ PartClassi คือ aggraggrname_wholeclassi_partclassi และผลคูณคาร์ทีเซียนระหว่างตัวแปรเซตของแอ็บสแตร็คแมชชีน BasicWholeClassi และ BasicPartClassi นั่นคือ BASICWHOLECLASSi × BASICPARTCLASSi

จะได้ว่า aggraggrname_wholeclassi_partclassi ⊂

BASICWHOLECLASSi × BASICPARTCLASSi .....(1)

กำหนด dom(aggraggrname_wholeclassi_partclassi) = basicwholeclassi.....(2)

และ ran(aggraggrname_wholeclassi_partclassi) = basicpartclassi.....(3)

กำหนดตัวแปร xx และ yy ภายใต้ตัวบ่งปริมาณ"ทั้งหมด" นั่นคือ ∀ (xx,yy)

ทำการสร้างเงื่อนไขของตัวแปร xx และ yy คือ

xx ∈ dom(aggraggrname_wholeclassi_partclassi) ∧

yy ∈ ran(aggraggrname_wholeclassi_partclassi) ภายใต้ตัวบ่งปริมาณทั้งหมดของตัวแปร xx และ yy นั่นคือ

∀ (xx,yy) (xx ∈ dom(aggraggrname_wholeclassi_partclassi) ∧

yy ∈ ran(aggraggrname_wholeclassi_partclassi))

ทำการกำหนดจำนวนเซตของตัวแปรของความสัมพันธ์แอกริเกชันระหว่าง WholeClassi และ PartClassi หรือ aggraggrname_wholeclassi_partclassi ของตัวแปร xx และ yy เพื่อระบุ

มัลติพลิซิตีของ WholeClassi และ PartClassi นั่นคือ

card((aggraggrname_wholeclassi_partclassi)[{xx}]) แทนมัลติพลิซิตีของ PartClassi นั่นคือ card((aggraggrname_wholeclassi_partclassi)[{xx}]) = 1 และ

card((aggraggrname_wholeclassi_partclassi)$^{-1}$[{yy}]) แทนมัลติพลิซิตีของ WholeClassi นั่นคือ

card((aggraggrname_wholeclassi_partclassi)$^{-1}$[{yy}]) ≥ 0 ∧

card((aggraggrname_wholeclassi_partclassi)$^{-1}$[{yy}]) ≤ 1

โดยมัลติพลิซิตีของ WholeClassi และ PartCiassi จะเป็นผลจากการระบุเงื่อนไขของตัวแปร xx และ yy ภายใต้ภายใต้ตัวปงปริมาณทั้งหมดของตัวแปร xx และ yy นั่นคือ

∀ (xx,yy) (((xx ∈ dom(aggraggrname_wholeclassi_partclassi) ∧

(yy ∈ ran(aggraggrname_wholeclassi_partclassi))

→ card((aggraggrname_wholeclassi_partclassi)[{xx}]) ≥ 1 ∧

card((aggraggrname_wholeclassi_partclassi)$^{-1}$[{yy}]) ≥ 0 ∧

card((aggraggrname_wholeclassi_partclassi)$^{-1}$[{yy}]) ≤ 1).....(4)

นำ (1) ∧ (2) ∧ (3) ∧ (4) จะได้ว่า

aggraggrname_wholeclassi_partclassi ⊂ BASICWHOLECLASSi × BASICPARTCLASSi ∧

dom(aggraggrname_wholeclassi_partclassi) = basicwholeclassi ∧

ran(aggraggrname_wholeclassi_partclassi) = basicpartclassi ∧

∀ (xx,yy) (((xx ∈ dom(aggraggrname_wholeclassi_partclassi) ∧

(yy ∈ ran(aggraggrname_wholeclassi_partclassi))

→ card((aggraggrname_wholeclassi_partclassi)[{xx}]) ≥ 1 ∧

card((aggraggrname_wholeclassi_partclassi)$^{-1}$[{yy}]) ≥ 0 ∧

card((aggraggrname_wholeclassi_partclassi)$^{-1}$[{yy}]) ≤ 1).....(4)

ในอนุประโยค INVARIANT

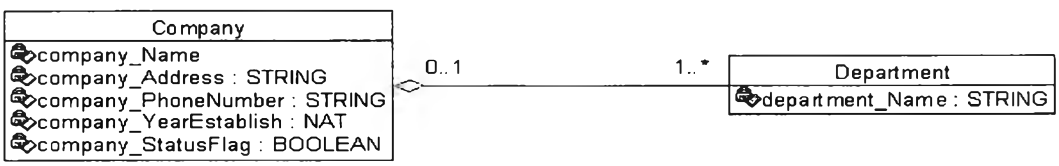
R5-7) กำหนดอนุประโยค INITIALISATION

R5-8) กำหนดค่าเริ่มต้นของเซตของตัวแปรของความสัมพันธ์แอกริเกชันระหว่าง WholeClassi และ PartClassj คือ aggraggrname_wholeclassi_partclassi := ϕ ในอนุประโยค INITIALISATION

R5-9) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี AggrAggrName_WholeClassi_PartClassi

R5-10) ระบุแอ็บสแตร็คแมชชีนบี AggrAggrName_WholeClassi_PartClassi ในอนุประโยค USES ของแอ็บสแตร็คแมชชีนบี WholeClassi และ PartClassi

ตัวอย่างของการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชัน โดยที่คลาส Company ซึ่งเป็นคลาสหลัก จะมีความสัมพันธ์แอกริเกชันกับคลาส Department ซึ่งเป็นคลาสย่อย นั่นคือ คลาส Company แต่ละคลาสสามารถมีความสัมพันธ์แบบแอกริเกชันกับคลาส Department ได้ตั้งแต่หนึ่งขึ้นไป หรือ 1...* ในทำนองเดียวกันคลาส Department แต่ละคลาสสามารถมีความสัมพันธ์แบบแอกริเกชันกับคลาส Company ได้เพียงตั้งแต่ศูนย์ แต่ไม่เกินหนึ่ง หรือ 0...1 แสดงได้ดังรูปที่ 3.31



รูปที่ 3.31 ตัวอย่างความสัมพันธ์แอกริเกชันระหว่างคลาส Company กับคลาส Department

พิจารณาจากรูปที่ 3.31 ในแอ็บสแตร็คแมชชีนบี BasicCompany จะทำการระบุตัวแปรรหัสอ้างถึงแอ็บสแตร็คแมชชีนบี BasicDepartment เพื่อให้แอ็บสแตร็คแมชชีนบี BasicCompany ทำการเก็บตัวแปรรหัสอ้างถึงของแอ็บสแตร็คแมชชีนบี BasicDepartment นั่นคือคลาส Company จะสามารถอ้างถึงคลาส Department ได้ แสดงแอ็บสแตร็คแมชชีนบี Company แอ็บสแตร็คแมชชีนบี BasicCompany แอ็บสแตร็คแมชชีนบี Department แอ็บสแตร็คแมชชีนบี BasicDepartment และแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาส Company กับคลาส Department โดยที่แอ็บสแตร็คแมชชีน Company และแอ็บสแตร็คแมชชีน Department จะระบุถึงแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่าง Company และ Department ในอนุประโยค USES จะแสดงได้ดังรูปที่ 3.32 ถึงรูปที่ 3.36

MACHINE
Company
USES
BasicCompany,
Aggr_Company_Department
END

รูปที่ 3.32 แอ็บสแตร็คแมชชีนบี Company

MACHINE
BasicCompany
...
VARIABLES
...,
RefDepartmentID
INVARIANT
... ∧
RefDepartmentID ∈ STRING
INITIALISATION
... ||
RefDepartmentID := null
END

รูปที่ 3.33 แอ็บสแตร็คแมชชีนบี BasicCompany

MACHINE
Department
USES
BasicDepartment,
Aggr_Company_Department
END

รูปที่ 3.34 แอ็บสแตร็คแมชชีนบี Department

MACHINE
BasicDepartment
...
END

รูปที่ 3.35 แอ็บสแตร็คแมชชีนบี BasicDepartment

MACHINE
Aggr_Company_Department
USES
BasicCompany, BasicDepartment
VARIABLES
aggr_company_department
INVARIANT
aggr_company_department ⊂ BASICCOMPANY × BASICDEPARTMENT ∧
dom(aggr_company_department) = basiccompany ∧
ran(aggr_company_department) = basicdepartment ∧
∀ (xx,yy).(((xx ∈ dom(aggr_company_department)) ∧
(yy ∈ ran(aggr_company_department)))
→ card((aggr_company_department)[{xx}]) ≥ 1 ∧
card((aggr_company_department)$^{-1}$[{yy}]) ≥ 0 ∧
card((aggr_company_department)$^{-1}$[{yy}]) ≤ 1)
INITIALISATION
aggr_company_department := ϕ
END

รูปที่ 3.36 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาส Company กับคลาส Department

กฎข้อที่ 6

การสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสหลักกับคลาสย่อย

ในกฎข้อนี้จะทำการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสหลักกับคลาสย่อย โดยที่ทุก ๆ ความสัมพันธ์คอมโพสิชันจะนำมาสร้างเป็นแอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสหลักกับคลาสย่อย นั่นคือ

จากความสัมพันธ์คอมโพสิชันระหว่างคลาสหลักกับคลาสย่อยในแผนภาพคลาส ซึ่งคลาสหลักและคลาสย่อย มีความสัมพันธ์คอมโพสิชัน โดยมีชื่อของความสัมพันธ์คือ CompoName แสดงได้ดังรูปที่ 3.37

รูปที่ 3.37 รูปแบบความสัมพันธ์คอมโพสิชันระหว่างคลาสหลักกับคลาสย่อย

จากรูปแบบของความสัมพันธ์คอมโพสิชันระหว่างคลาสหลักกับคลาสย่อย คลาสหลักจะทำการระบุรหัสอ้างถึงคลาสย่อยแต่ละคลาส นั่นคือ คลาสหลักจะทำการเก็บรหัสของคลาสย่อย เพื่อให้คลาสหลักสามารถอ้างถึงคลาสย่อยได้ ซึ่งการระบุรหัสอ้างถึงนั้น จะทำการระบุตัวแปรรหัสอ้างถึงคลาสย่อยในอนุประโยค VARIABLES สร้างความสัมพันธ์ที่มีเงื่อนไขที่เป็นจริงของตัวแปรรหัสอ้างถึงคลาสย่อยในอนุประโยค INVARIANT และระบุค่าเริ่มต้นตัวแปรรหัสอ้างถึงคลาสย่อยในอนุประโยค INITIALISATION ในแอ็บสแตร็คแมชชีนบีเบซิคคลาสหลักเช่นเดียวกับรูปแบบของความสัมพันธ์แอกริเกชันระหว่างคลาสหลักกับคลาสย่อย โดยที่ แอ็บสแตร็คแมชชีนบี WholeClass และแอ็บสแตร็คแมชชีนบี PartClass จะระบุถึงแอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสหลักและคลาสย่อยนั่นคือ CompoCompoName_WholeClass_PartClass ในอนุประโยค USES

จากรูปที่ 3.37 สามารถนำมาสร้างเป็นแอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสหลักและคลาสย่อย แสดงได้ดังรูปที่ 3.38

MACHINE
CompoCompoName_WholeClass_PartClass
USES
BasicWholeClass, BasicPartClass
VARIABLES
compocomponame_wholeclass_partclass
INVARIANT
compocomponame_wholeclass_partclass ⊂ BASICWHOLECLASS × BASICPARTCLASS ∧
dom(compocomponame_wholeclass_partclass) = basicwholeclass ∧
ran(compocomponame_wholeclass_partclass) = basicpartclass ∧
∀ (xx,yy).(((xx ∈ dom(compocomponame_wholeclass_partclass)) ∧
(yy ∈ ran(compocomponame_wholeclass_partclass)))
→ card((compocomponame_wholeclass_partclass)[{xx}]) ≥ 1 ∧
card((compocomponame_wholeclass_partclass)$^{-1}$ [{yy}]) = 1)
INITIALISATION
compocomponame_wholeclass_partclass := ϕ
END

รูปที่ 3.38 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสหลักกับคลาสย่อย

โดยที่ compocomponame_wholeclass_partclass

คือ ตัวแปรในแอ็บสแตร็คแมชชีนบีที่แสดงถึงความสัมพันธ์คอมโพสิชันโดยมีคำนำหน้า compo ติดกับชื่อของความสัมพันธ์ CompoName แล้วตามด้วยชื่อของ WholeClass และ PartClass โดยชื่อของความสัมพันธ์คอมโพสิชันและคลาสทั้ง 2 ต้องใช้อักษรภาษาอังกฤษตัวพิมพ์เล็ก

dom และ ran

คือ การระบุโดเมน และพิสัยของความสัมพันธ์คอมโพสิชัน

xx และ yy

คือ ตัวแปรที่เป็นสมาชิกของโดเมน และพิสัยของความสัมพันธ์คอมโพสิชัน

card((compocompo_wholeclass_partclass) [{xx}])

คือ จำนวนสมาชิกของโดเมนจากความสัมพันธ์คอมโพสิชันระหว่าง WholeClass และ PartClass

card((compocompo_wholeclass_partclass)$^{-1}$ [{yy}])

คือ จำนวนสมาชิกของสวนผกผันพิสัยจากความสัมพันธ์คอมโพสิชันระหว่าง WholeClass และ PartClass

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 6 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 6

สำหรับ WholeClassi และ PartClassi ใด ๆ ในแผนภาพคลาส โดยมีความสัมพันธ์คอมโพสิชัน CompoName ระหว่าง WholeClassi และ PartClassi

จากข้อความข้างต้น จะทำการสร้างแอ็บสแตร็คแมชชีนบี CompoCompoName_WholeClassi_PartClassi ได้ดังนี้

R6-1) กำหนดอนุประโยค MACHINE

R6-2) กำหนดชื่อของแอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่าง WholeClassi และ PartClassi คือ CompoCompoName_WholeClassi_PartClassi ในอนุประโยค MACHINE

R6-3) กำหนดอนุประโยค USES

R6-4) ระบุชื่อของแอ็บสแตร็คแมชชีนบี BasicWholeClassi และ BasicPartClassi ในอนุประโยค USES

R6-5) จากข้อที่ R6-1 กำหนดเซตของตัวแปรของความสัมพันธ์คอมโพสิชันระหว่าง WholeClassi และ PartClassi คือ compocomponame_wholeclassi_partclassi ในอนุประโยค VARIABLES

R6-6) กำหนดความสัมพันธ์ระหว่างเซตของตัวแปรของความสัมพันธ์คอมโพสิชันระหว่าง WholeClassi และ PartClassi คือ compocomponame_wholeclassi_partclassi และผลคูณคาร์ทีเซียนระหว่างตัวแปรเซตของแอ็บสแตร็คแมชชีน BasicWholeClassi และ BasicPartClassi นั่นคือ BASICWHOLECLASSi × BASICPARTCLASSi

จะได้ว่า compocomponame_wholeclassi_partclassi ⊂

BASICWHOLECLASSi × BASICPARTCLASSi .....(1)

กำหนด dom(compocomponame_wholeclassi_partclassi) = basicwholeclassi.....(2)

และ ran(compocomponame_wholeclassi_partclassi) = basicpartclassi.....(3)

กำหนดตัวแปร xx และ yy ภายใต้ตัวบ่งปริมาณ"ทั้งหมด" นั่นคือ ∀ (xx,yy)

ทำการสร้างเงื่อนไขของตัวแปร xx และ yy คือ

xx ∈ dom(compocomponame_wholeclassi_partclassi) ∧

yy ∈ ran(compocomponame_wholeclassi_partclassi) ภายใต้ตัวบ่งปริมาณทั้งหมดของตัวแปร xx และ yy นั่นคือ

∀ (xx,yy) (xx ∈ dom(compocomponame_wholeclassi_partclassi) ∧

yy ∈ ran(compocomponame_wholeclassi_partclassi))

ทำการกำหนดจำนวนเซตของตัวแปรของความสัมพันธ์คอมโพสิชันระหว่าง WholeClassi และ PartClassi หรือ compocomponame_wholeclassi_partclassi ของตัวแปร xx และ yy เพื่อระบุมัลติพลิซิตีของ WholeClassi และ PartCiassi นั่นคือ

card((compocomponame_wholeclassi_partclassi)[{xx}]) แทนมัลติพลิซิตีของ PartClassi นั่นคือ card((compocomponame_wholeclassi_partclassi)[{xx}]) ≥ 1

และ card((compocomponame_wholeclassi_partclassi)$^{-1}$[{yy}]) แทนมัลติพลิซิตีของ WholeClassi นั่นคือ card((compocomponame_wholeclassi_partclassi)$^{-1}$[{yy}]) = 1

โดยมัลติพลิซิตีของ WholeClassi และ PartCiassi จะเป็นผลจากการระบุเงื่อนไขของตัวแปร xx และ yy ภายใต้ภายใต้ตัวบ่งปริมาณทั้งหมดของตัวแปร xx และ yy นั่นคือ

$\forall$ (xx,yy) (((xx $\in$ dom(compocomponame_wholeclassi_partclassi) $\wedge$

(yy $\in$ ran(compocomponame_wholeclassi_partclassi))

$\rightarrow$ card((compocomponame_wholeclassi_partclassi)[{xx}]) $\geq$ 1 $\wedge$

card((compocomponame_wholeclassi_partclassi)$^{-1}$[{yy}]) = 1).....(4)

นำ (1) $\wedge$ (2) $\wedge$ (3) $\wedge$ (4) จะได้ว่า

compocomponame_wholeclassi_partclassi $\subset$ BASICWHOLECLASSi $\times$ BASICPARTCLASSi $\wedge$

dom(compocomponame_wholeclassi_partclassi) = basicwholeclassi $\wedge$

ran(compocomponame_wholeclassi_partclassi) = basicpartclassi $\wedge$

$\forall$ (xx,yy) (((xx $\in$ dom(compocomponame_wholeclassi_partclassi) $\wedge$

(yy $\in$ ran(compocomponame_wholeclassi_partclassi))

$\rightarrow$ card((compocomponame_wholeclassi_partclassi)[{xx}]) $\geq$ 1 $\wedge$

card((compocomponame_wholeclassi_partclassi)$^{-1}$[{yy}]) = 1).....(4)

ในอนุประโยค INVARIANT

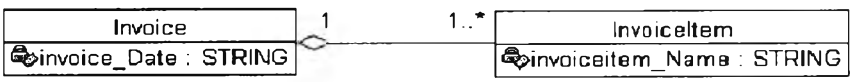
R6-7) กำหนดอนุประโยค INITIALISATION

R6-8) กำหนดค่าเริ่มต้นของเซตของตัวแปรของความสัมพันธ์คอมโพสิชันระหว่าง WholeClassi และ PartCiassi คือ compocomponame_wholeclassi_partclassi := ϕ ในอนุประโยค INITIALISATION

R6-9) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี CompocompoName_WholeClassi_PartClassi

R6-10) ระบุแอ็บสแตร็คแมชชีนบี CompocompoName_WholeClassi_PartClassi ในอนุประโยค USES ของแอ็บสแตร็คแมชชีนบี WholeClassi และ PartCiassi

ตัวอย่างการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชัน คือ คลาส Invoice ซึ่งเป็นคลาสหลัก จะมีความสัมพันธ์คอมโพสิชันกับคลาส InvoiceItem ซึ่งเป็นคลาสย่อย นั่นคือ คลาส Invoice แต่ละคลาสสามารถมีความสัมพันธ์แบบคอมโพสิชันกับคลาส InvoiceItem ได้ตั้งแต่หนึ่งขึ้นไป หรือ 1...* ในทำนองเดียวกันคลาส InvoiceItem โดยแต่ละคลาสสามารถมีความสัมพันธ์แบบคอมโพสิชันกับกับคลาส Invoice ได้เพียงหนึ่ง หรือ 1 แสดงได้ดังรูปที่ 3.39



รูปที่ 3.39 ตัวอย่างความสัมพันธ์คอมโพสิชันชันระหว่างคลาส Invoice กับคลาส InvoiceItem

จากรูปที่ 3.39 แอ็บสแตร็คแมชชีนบี BasicInvoice ทำการระบุตัวแปรรหัสอ้างถึงแอ็บสแตร็คแมชชีนบี BasicInvoiceItem เพื่อให้แอ็บสแตร็คแมชชีนบี BasicInvoice สามารถทำการเก็บตัวแปรรหัสอ้างถึงของแอ็บสแตร็คแมชชีนบี BasicInvoiceItem นั่นคือคลาส Invoice สามารถทำการอ้างถึงคลาส InvoiceItem ได้ แสดงแอ็บสแตร็คแมชชีนบี Invoice แอ็บสแตร็คแมชชีนบี BasicInvoice แอ็บสแตร็คแมชชีนบี InvoiceItem แอ็บสแตร็คแมชชีนบี BasicInvoiceItem และแอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาส Invoice กับคลาส InvoiceItem โดยที่แอ็บสแตร็คแมชชีนบี Invoice และแอ็บสแตร็คแมชชีนบี InvoiceItem จะระบุถึงแอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่าง Invoice กับ InvoiceItem ในอนุประโยค USES จะแสดงได้ดังรูปที่ 3.40 ถึงรูปที่ 3.44

```
MACHINE
Invoice
USES
BasicInvoice,
Compo_Invoice_InvoiceItem
END
```

รูปที่ 3.40 แอ็บสแตร็คแมชชีนบี Invoice

```
MACHINE
BasicInvoice
...
VARIABLES
...,
RefInvoiceItemID
INVARIANT
... ∧
RefInvoiceItemID ∈ STRING
INITIALISATION
...||
RefInvoiceItemID := null
END
```

รูปที่ 3.41 แอ็บสแตร็คแมชชีนบี BasicInvoice

MACHINE
InvoiceItem
USES
BasicInvoiceItem,
Compo_Invoice_InvoiceItem

END

รูปที่ 3.42 แอ็บสแตร็คแมชชีนบี InvoiceItem

MACHINE
BasicInvoiceItem
...
END

รูปที่ 3.43 แอ็บสแตร็คแมชชีนบี BasicInvoiceItem

MACHINE
Compo_Invoice_InvoiceItem
USES
BasicInvoice, BasicInvoiceItem
VARIABLES
compo_invoice_invoiceitem
INVARIANTS
compo_invoice_invoiceitem $\subset$ BASICINVOICE $\times$ BASICINVOICEITEM $\wedge$
dom(compo_invoice_invoiceitem) = basicinvoice $\wedge$
ran(compo_invoice_invoiceitem) = basicinvoiceitem $\wedge$
$\forall$ (xx,yy).(((xx $\in$ dom(compo_invoice_invoiceitem)) $\wedge$
(yy $\in$ ran(compo_invoice_invoiceitem)))
$\rightarrow$ card((compo_invoice_invoiceitem)[{xx}]) $\geq$ 1 $\wedge$
card((compo_invoice_invoiceitem)$^{-1}$[{yy}]) = 1)
INITIALISATION
compo_invoice_invoiceitem := ϕ
END

รูปที่ 3.44 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาส Invoice กับคลาส InvoiceItem

3.2.3 การสร้างแอ็บสแตร็คแมชชีนบีซับคลาสจากความสัมพันธ์เจเนอรัลไลเซชัน

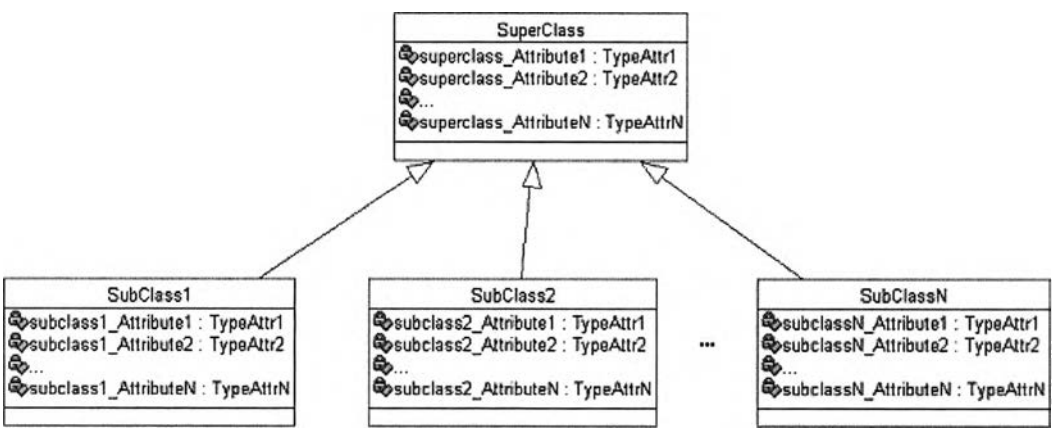
ความสัมพันธ์เจเนอรัลไลเซชันของคลาส จะเป็นการถ่ายทอดคุณลักษณะทั้งหมดที่มีของซูเปอร์คลาสไปยังซับคลาส ทำให้ซับคลาสที่ได้รับการถ่ายทอดมาจากซูเปอร์คลาสสามารถมีคุณลักษณะเฉพาะที่เจาะจงมากกว่าซูเปอร์คลาส ซึ่งการสร้างแอ็บสแตร็คแมชชีนบีจากความสัมพันธ์เจเนอรัลไลเซชันของคลาสนั้น จะเป็นการสร้างแอ็บสแตร็คแมชชีนบีเบซิคซับคลาสที่ได้รับถ่ายทอดคุณลักษณะทั้งหมดมาจากซูเปอร์คลาส โดยที่แอ็บสแตร็คแมชชีนบีเบซิคซับคลาส จะต้องระบุคุณลักษณะทั้งหมดมาจากซูเปอร์คลาส

พิจารณาความสัมพันธ์ทั้งหมด ได้แก่ ความสัมพันธ์แอสโซซิเอชัน ความสัมพันธ์แอกริเกชัน และความสัมพันธ์คอมโพสิชันระหว่างซูเปอร์คลาสกับคลาสอื่น ๆ จะสามารถนำมาสร้างเป็นแอ็บสแตร็คแมชชีนบีของความสัมพันธ์โดยปริยาย (Implicit Relation) ของซับคลาสที่มาจากความสัมพันธ์ระหว่างซูเปอร์คลาสกับคลาสอื่น ๆ ได้ ในงานวิจัยนี้ได้นำเสนอกฎการแปลงความสัมพันธ์เจเนอรัลไลเซชันของแผนภาพคลาสไปเป็นแอ็บสแตร็คแมชชีนบี ดังนี้

กฎข้อที่ 7

การสร้างแอ็บสแตร็คแมชชีนบีซับคลาสและเบซิคซับคลาสที่รับการถ่ายทอดคุณลักษณะทั้งหมดมาจากซูเปอร์คลาส

จากรูปแบบของความสัมพันธ์เจเนอรัลไลเซชันของคลาสในแผนภาพคลาส นำมาสร้างเป็นแอ็บสแตร็คแมชชีนบีซับคลาส กล่าวคือ แอ็บสแตร็คแมชชีนบีเบซิคซับคลาสที่สร้างขึ้นมานั้น จะต้องทำการระบุคุณลักษณะทั้งหมดมาจากซูเปอร์คลาส แสดงได้ดังรูปที่ 3.45



รูปที่ 3.45 รูปแบบความสัมพันธ์เจนเนอรัลไลเซชันของซูเปอร์คลาสและซับคลาสหลาย ๆ คลาส

ซูเปอร์คลาส SuperClass ใด ๆ จะมีซับคลาส SubClass1, SubClass2, ...,SubClassN ในแผนภาพคลาส สามารถนำมาสร้างเป็นแอ็บสแตร็คแมชชีนบีจากความสัมพันธ์เจเนอรัลไลเซชันประกอบด้วยแอ็บสแตร็คแมชชีนบีซูเปอร์คลาส (SuperClass) แอ็บสแตร็คแมชชีนบีเบซิคซูเปอร์คลาส (BasicSuperClass) ที่ระบุตัวแปรคุณลักษณะทั้งหมดของซูเปอร์คลาสและชนิดของคุณลักษณะทั้งหมดได้แก่ superclass_Attribute1, superclass_Attribute2, ..., superclass_AttributeN และ TypeAttr1, TypeAttr2, ..., TypeAttrN ตามลำดับ แอ็บสแตร็คแมชชีนบีซับคลาส (SubClass) ทั้งหมด ได้แก่ SubClass1, SubClass2, ..., SubClassN และแอ็บสแตร็คแมชชีนบีเบซิคซับคลาส (BasicSubClass) ทั้งหมด ได้แก่ BasicSubClass1, BasicSubClass2, ..., BasicSubClassN ที่ระบุตัวแปรคุณลักษณะเฉพาะทั้งหมดของซับคลาส ได้แก่ subclass1_Attribute1, subclass1_Attribute2, ..., subclass1_AttributeN, subclass2_Attribute1,

subclass2_Attribute2, ..., subclass2_AttributeN, subclassN_Attribute1, subclassN_Attribute2, ..., subclassN_AttributeN ซึ่งมีชนิดของคุณลักษณะทั้งหมดของซับคลาส ได้แก่ TypeAttr1, TypeAttr2, ..., TypeAttrN โดยแอ็บสแตร็คแมชชีนบีแมชชีนบี BasicSubClass ทั้งหมด จะระบุตัวแปรคุณลักษณะทั้งหมดของซูเปอร์คลาส แสดงได้ดังรูปที่ 3.46 ถึงรูปที่ 3.49

```
MACHINE
SuperClass
USES
BasicSuperClass
...
END
```

รูปที่ 3.46 แอ็บสแตร็คแมชชีนบี SuperClass

```
MACHINE
BasicSuperClass
...
VARIABLES
superclass_Attribute1,
supeclass_Attribute2,
...,
superclass_AttributeN
INVARIANT
superclass_Attribute1 ∈ TypeAttr1 ∧
superclass_Attribute2 ∈ TypeAttr2 ∧
... ∧
superclass_AttributeN ∈ TypeAttrN
...
END
```

รูปที่ 3.47 แอ็บสแตร็คแมชชีนบี BasicSuperClass

```
MACHINE
SubClass1
USES
BasicSubClass1
...
END
```

```
MACHINE
SubClass2
USES
BasicSubClass2
...
END
```

...

MACHINE
SubClassN
USES
BasicSubClassN
...
END

รูปที่ 3.48 แอ็บสแตร็คแมชชีนบี SubClass ทั้งหมด

MACHINE
BasicSubClass1
...
VARIABLES
subclass1_Attribute1,
subclass1_Attribute2,
...,
subclass1_AttributeN,
superclass_Attribute1,
superclass_Attribute2,
...,
superclass_AttributeN
INVARIANT
subclass1_Attribute1 ∈ TypeAttr1 ∧
subclass1_Attribute2 ∈ TypeAttr2 ∧
... ∧
subclass1_AttributeN ∈ TypeAttrN ∧
superclass_Attribute1 ∈ *TypeAttr1* ∧
superclass_Attribute2 ∈ *TypeAttr2* ∧
... ∧
superclass_AttributeN ∈ *TypeAttrN*
...
END

MACHINE
BasicSubClass2
...
VARIABLES
subclass2_Attribute1,
subclass2_Attribute2,
...,
subclass2_AttributeN,
superclass_Attribute1,
superclass_Attribute2,
...,
superclass_AttributeN
INVARIANT
subclass2_Attribute1 ∈ TypeAttr1 ∧
subclass2_Attribute2 ∈ TypeAttr2 ∧
... ∧
subclass2_AttributeN ∈ TypeAttrN ∧
superclass_Attribute1 ∈ *TypeAttr1* ∧
superclass_Attribute2 ∈ *TypeAttr2* ∧
... ∧
superclass_AttributeN ∈ *TypeAttrN*
...
END

...

MACHINE
BasicSubClassN
...
VARIABLES
subclassN_Attribute1,
subclassN_Attribute2,
...,
subclassN_AttributeN,
superclass_Attribute1,
superclass_Attribute2,
...,
superclass_AttributeN
INVARIANT
subclassN_Attribute1 ∈ TypeAttr1 ∧
subclassN_Attribute2 ∈ TypeAttr2 ∧
... ∧
subclassN_AttributeN ∈ TypeAttrN ∧
superclass_Attribute1 ∈ *TypeAttr1* ∧
superclass_Attribute2 ∈ *TypeAttr2* ∧
... ∧
superclass_AttributeN ∈ *TypeAttrN*
...
END

รูปที่ 3.49 แอ็บสแตร็คแมชชีนบี BasicSubClass ทั้งหมด

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 7 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 7

สำหรับ SuperClassi ใด ๆ ในแผนภาพคลาส โดยที่ SuperClassi จะมีคุณลักษณะทั้งหมด ได้แก่ superclassi_Attribute1, superclassi_Attribute2, ..., superclassi_AttributeN ซึ่ง SubClassi จะมีคุณลักษณะทั้งหมด ได้แก่ subclassi_Attribute1, subclassi_Attribute2, ..., subclassi_AttributeN ดังนั้น SuperClassi ใด ๆ จะถ่ายทอดคุณลักษณะทั้งหมดนี้ให้กับ SubClassi นั่นคือ SubClassi จะมีคุณลักษณะทั้งหมดของ SuperClassi ได้แก่ superclassi_Attribute1, superclassi_Attribute2, ..., superclassi_AttributeN

จากข้อความข้างต้น จะทำการสร้างแอ็บสแตร็คแมชชีนบี BasicSuperClassi แอ็บสแตร็คแมชชีนบี SuperClassi แอ็บสแตร็คแมชชีนบี BasicSubClassi และแอ็บสแตร็คแมชชีนบี SubClassi ได้ดังนี้

สำหรับแอ็บสแตร็คแมชชีนบี BasicSuperClassi ทำได้ดังนี้

R7-1) กำหนดอนุประโยค MACHINE

R7-2) กำหนดชื่อของแอ็บสแตร็คแมชชีนบี คือ BasicSuperClassi ในอนุประโยค MACHINE

R7-3) กำหนดอนุประโยค SETS

R7-4) กำหนดเซตของแอ็บสแตร็คแมชชีนบี BasicSuperClassi คือ BASICSUPERCLASSi ในอนุประโยค SETS

R7-5) กำหนดอนุประโยค VARIABLES

R7-6) กำหนดตัวแปรเซตของแอ็บสแตร็คแมชชีนบี BasicSuperClassi คือ basicsuperclassi และตัวแปรคุณลักษณะทั้งหมด คือ superclassi_Attribute1, superclassi_Attribute2, ..., superclassi_AttributeN ในอนุประโยค VARIABLES

R7-7) กำหนดอนุประโยค INVARIANT

R7-8) กำหนดความสัมพันธ์ระหว่างตัวแปรเซตกับเซตของ SuperClassi ในแอ็บสแตร็คแมชชีนบี BasicSuperClassi นั่นคือ basicsuperclassi $\subset$ BASICSUPERCLASSi ....(SupInv1)
และกำหนดความสัมพันธ์ระหว่างตัวแปรคุณลักษณะและชนิดของคุณลักษณะทั้งหมด นั่นคือ superclassi_Attribute1 $\in$ TypeAttr1.....(SupInv2)
superclassi_Attribute2 $\in$ TypeAttr2.....(SupInv3)

...

superclassi_AttributeN $\in$ TypeAttrN.....(SupInvN+1)
นำความสัมพันธ์ทั้งหมด ได้แก่ (SupInv1), (SupInv2), ..., (SupInvN+1) มาเชื่อมด้วยตัวเชื่อมตรรกคณิตศาสตร์ คือ ตัวเชื่อม "และ" นั่นคือ $\wedge$ จะได้ความสัมพันธ์ทั้งหมด คือ (SupInv1) $\wedge$ (SupInv2) $\wedge$ ... $\wedge$ (SupInvN+1) ในอนุประโยค INVARIANT

R7-9) กำหนดอนุประโยค INITIALISATION

R7-10) กำหนดค่าเริ่มต้นของตัวแปรเซต นั่นคือ basicsuperclassi := ϕ .....(SupInit1)
และกำหนดค่าเริ่มให้ตัวแปรคุณสักษณะทั้งหมด นั่นคือ
superclassi_Attribute1 := InitialValue .....(SupInit2)
superclassi_Attribute2 := InitialValue .....(SupInit3)

...

superclassi_AttributeN := InitialValue .....(SupInitN+1)
นำความสัมพันธ์ทั้งหมด ได้แก่ (SupInit1), (SupInit2), ..., (SupInitN+1) มาเชื่อมด้วยเครื่องหมายการกำหนดค่าแบบขนาน นั่นคือ || จะได้ค่าเริ่มต้นของตัวแปรเซตและตัวแปรคุณลักษณะทั้งหมด คือ (SupInit1) || (SupInit2) || ... || (SupInitN+1) ในอนุประโยค

INITIALISATION

R7-11) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี BasicSuperClass

สำหรับแอ็บสแตร็คแมชชีนบี SuperClassi ทำได้ดังนี้

R7-12) กำหนดอนุประโยค MACHINE

R7-13) กำหนดชื่อของแอ็บสแตร็คแมชชีนบี คือ SuperClassi ในอนุประโยค MACHINE

R7-14) กำหนดอนุประโยค USES

R7-15) ระบุชื่อของแอ็บสแตร็คแมชชีนบี คือ BasicSuperClassi ในอนุประโยค USES

R7-16) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี SuperClassi

สำหรับแอ็บสแตร็คแมชชีนบี BasicSubClassi ทำได้ดังนี้

R7-17) กำหนดอนุประโยค MACHINE

R7-18) กำหนดชื่อของแอ็บสแตร็คแมชชีนบี คือ BasicSubClassi ในอนุประโยค MACHINE

R7-19) กำหนดอนุประโยค SETS

R7-20) กำหนดเซตของแอ็บสแตร็คแมชชีนบี BasicSubClassi คือ BASICSUBCLASSi ในอนุประโยค SETS

R7-21) กำหนดอนุประโยค VARIABLES

R7-22) กำหนดตัวแปรเซตของแอ็บสแตร็คแมชชีนบี BasicSubClassi คือ basicsubclassi และตัวแปรคุณลักษณะทั้งหมดของ SubClassi คือ subclassi_Attribute1, subclassi_Attribute2, ..., subclassi_AttributeN รวมทั้งตัวแปรคุณลักษณะทั้งหมดของ SuperClassi คือ superclassi_Attribute1, superclassi_Attribute2, ..., superclassi_AttributeN ในอนุประโยค VARIABLES

R7-23) กำหนดอนุประโยค INVARIANT

R7-24) กำหนดความสัมพันธ์ระหว่างตัวแปรเซตกับเซตของ SubClassi ในแอ็บสแตร็คแมชชีนบี BasicSubClassi นั่นคือ basicsubclassi $\subset$ BASICSUBCLASSi .....(SubInv1) และกำหนดความสัมพันธ์ระหว่างตัวแปรคุณลักษณะและชนิดของคุณลักษณะทั้งหมดของ SubClassi นั่นคือ

subclassi_Attribute1 $\in$ TypeAttr1.....(SubInv2)

subclassi_Attribute2 $\in$ TypeAttr2.....(SubInv3)

...

subclassi_AttributeN $\in$ TypeAttrN.....(SubInvN+1)

รวมทั้งกำหนดความสัมพันธ์ระหว่างตัวแปรคุณลักษณะ และชนิดของคุณลักษณะทั้งหมดของ SuperClassi นั่นคือ

superclassi_Attribute1 ∈ TypeAttr1.....(SupInv1)

superclassi_Attribute2 ∈ TypeAttr2.....(SupInv2)

...

superclassi_AttributeN ∈ TypeAttrN.....(SupInvN)

นำความสัมพันธ์ทั้งหมด ได้แก่ (SubInv1), (SubInv2), ..., (SubInvN+1), (SupInv1), (SupInv2), ..., (SupInvN) มาเชื่อมด้วยตัวเชื่อมตรรกคณิตศาสตร์ คือ ตัวเชื่อม "และ" นั่นคือ ∧ จะได้ความสัมพันธ์ทั้งหมด คือ

(SubInv1) ∧ (SubInv2) ∧ ... ∧ (SubInvN+1) ∧ (SupInv1) ∧ (SupInv2) ∧ ... ∧ (SupInvN) ในอนุประโยค INVARIANT

R7-25) กำหนดอนุประโยค INITIALISATION

R7-26) กำหนดค่าเริ่มต้นของตัวแปรเซต นั่นคือ basicsubclassi := ϕ .....(SubInit1)

และกำหนดค่าเริ่มให้ตัวแปรของคุณลักษณะทั้งหมดของ SubClass นั่นคือ

subclassi_Attribute1 := InitialValue .....(SubInit2)

subclassi_Attribute2 := InitialValue .....(SubInit3)

...

subclassi_AttributeN := InitialValue .....(SubInitN+1)

รวมทั้งกำหนดค่าเริ่มต้นของตัวแปรคุณลักษณะทั้งหมดของ SuperClass นั่นคือ

superclass_Attribute1 := InitialValue .....(SupInit1)

superclassi_Attribute2 := InitialValue .....(SupInit2)

...

superclassi_AttributeN := InitialValue .....(SupInitN)

นำความสัมพันธ์ทั้งหมด ได้แก่ (SubInit1), (SubInit2), ...,(SubInitN+1), (SupInit1), (SupInit2),..., (SupInitN)

มาเชื่อมด้วยเครื่องหมายการกำหนดค่าแบบขนาน นั่นคือ || จะได้ค่าเริ่มต้นของตัวแปรของเซตและตัวแปรของคุณลักษณะทั้งหมด คือ (SubInit1) || (SubInit2) || ... || (SubInitN+1) || (SupInit1) || (SupInit2) || ... || (SupInitN) ในอนุประโยค INITIALISATION

R7-27) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี BasicSubClassi

สำหรับแอ็บสแตร็คแมชชีนบี SubClassi **ทำได้ดังนี้**

R7-28) กำหนดอนุประโยค MACHINE

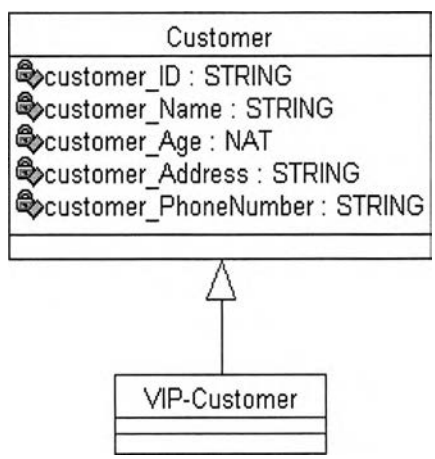
R7-29) กำหนดชื่อของแอ็บสแตร็คแมชชีนบี คือ SubClassi ในอนุประโยค MACHINE

R7-30) กำหนดอนุประโยค USES

R7-31) ระบุชื่อของแอ็บสแตร็คแมชชีนบี คือ BasicSubClassi ในอนุประโยค USES

R7-32) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี SubClassi

ตัวอย่างคลาส Customer ซึ่งเป็นซูเปอร์คลาส โดยถ่ายทอดคุณลักษณะทั้งหมดให้กับคลาส VIP-Customer ซึ่งเป็นซับคลาส แสดงได้ดังรูปที่ 3.50



รูปที่ 3.50 ตัวอย่างของคลาส Customer ที่ถ่ายทอดคุณลักษณะให้กับคลาส VIP-Customer

จากรูปที่ 3.50 จะแสดงตัวอย่างของความสัมพันธ์เจเนอรัลไลเซชัน โดยสามารถนำมาสร้างเป็นแอ็บสแตร็คแมชชีนบี Customer แอ็บสแตร็คแมชชีนบี BasicCustomer แอ็บสแตร็คแมชชีนบี VIP-Customer และแอ็บสแตร็คแมชชีนบี BasicVIP-Customer แสดงได้ดังรูปที่ 3.51 ถึงรูปที่ 3.54

```
MACHINE
Customer
USES
BasicCustomer
...
END
```

รูปที่ 3.51 แอ็บสแตร็คแมชชีนบี Customer ซึ่งเป็นซูเปอร์คลาส

MACHINE
BasicCustomer
...
SETS
BASICCUSTOMER
VARIABLES
basiccustomer,
customer_ID,
customer_Name,
customer_Age,
customer_Address,
customer_PhoneNumber,
customer_Status
INVARIANT
basiccustomer $\subset$ BASICCUSTOMER $\wedge$
customer_ID $\in$ STRING $\wedge$
customer_Name $\in$ STRING $\wedge$
customer_Age $\in$ NAT $\wedge$
customer_Address $\in$ STRING $\wedge$
customer_PhoneNumber $\in$ STRING $\wedge$
customer_Status $\in$ BOOLEAN
INITIALISATION
basiccustomer := ϕ ||
customer_ID := null ||
customer_Name := null ||
customer_Age := 0 ||
customer_Address := null ||
customer_PhoneNumber := null ||
customer_Status := FALSE
END

รูปที่ 3.52 แอ็บสแตร็คแมชชีนบี BasicCustomer ซึ่งเป็นซูเปอร์คลาส

MACHINE
VIP-Customer
USES
BasicVIP-Customer
...
END

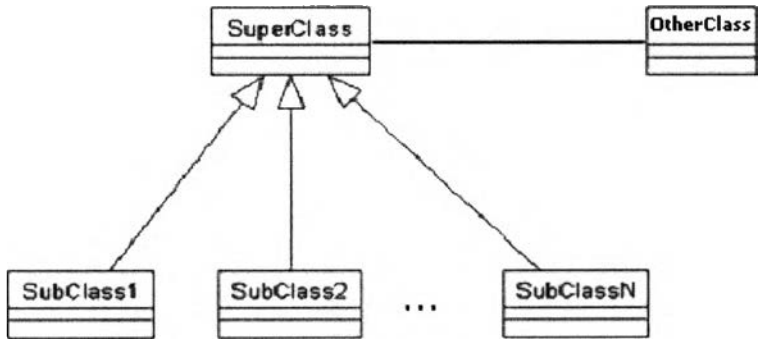
รูปที่ 3.53 แอ็บสแตร็คแมชชีนบี VIP-Customer ซึ่งเป็นซับคลาส

MACHINE
BasicVIP-Customer
...
SETS
BASICVIP-CUSTOMER
VARIABLES
basicvip-customer,
vip-customer_ID,
vip-customer_Name,
vip-customer_Age,
vip-customer_Address,
vip-customer_PhoneNumber,
vip-customer_Status
INVARIANT
basicvip-customer $\subset$ BASICVIP-CUSTOMER $\wedge$
vip-customer_ID $\in$ STRING $\wedge$
vip-customer_Name $\in$ STRING $\wedge$
vip-customer_Age $\in$ NAT $\wedge$
vip-customer_Address $\in$ STRING $\wedge$
vip-customer_PhoneNumber $\in$ STRING $\wedge$
vip-customer_Status $\in$ BOOLEAN
INITIALISATION
basicvip-customer := ϕ ||
vip-customer_ID := null ||
vip-customer_Name := null ||
vip-customer_Age := 0 ||
vip-customer_Address := null ||
vip-customer_PhoneNumber := null ||
vip-customer_Status := FALSE
END

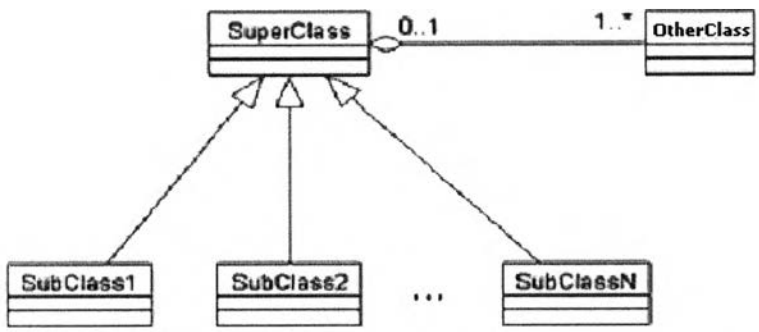
รูปที่ 3.54 แอ็บสแตร็คแมชชีนบี BasicVIP-Customer ซึ่งเป็นซับคลาส

กฎข้อที่ 8
การสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์โดยปริยายของซับคลาสจากความสัมพันธ์ระหว่างซูเปอร์คลาสกับคลาสอื่น ๆ

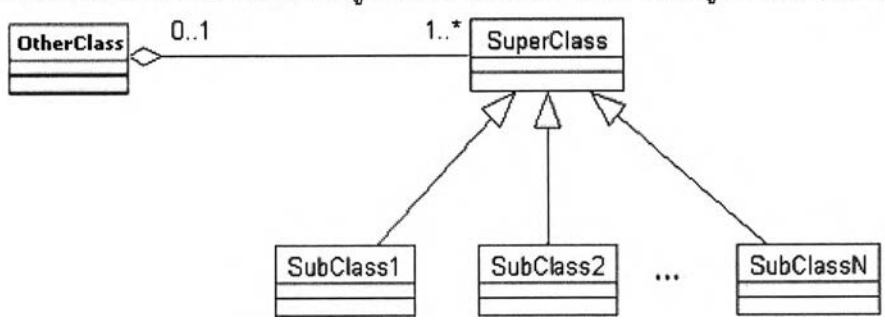
จากความสัมพันธ์ระหว่างคลาสในแผนภาพคลาส ได้แก่ ความสัมพันธ์แอสโซซิเอชัน ความสัมพันธ์แอกริเกชัน และความสัมพันธ์คอมโพสิชันระหว่างซูเปอร์คลาส (SuperClass) กับคลาสอื่น (OtherClass) โดยที่ซูเปอร์คลาสจะสามารถถ่ายทอดคุณลักษณะให้กับซับคลาสทั้งหมด ได้แก่ SubClass1, SubClass2, ..., SubClassN แสดงได้ดังรูปที่ 3.55 ถึงรูปที่ 3.59



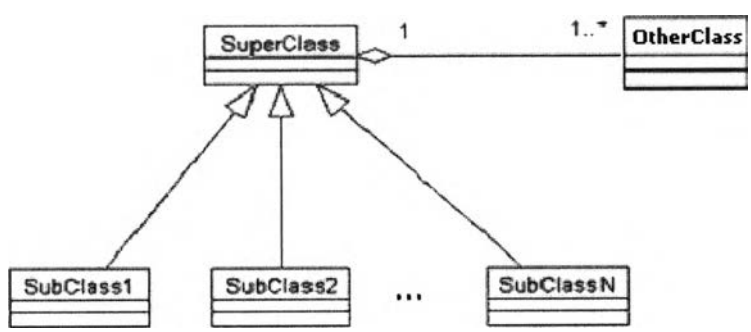
รูปที่ 3.55 ความสัมพันธ์แอสโซซิเอชันระหว่างซูเปอร์คลาสกับคลาสอื่น



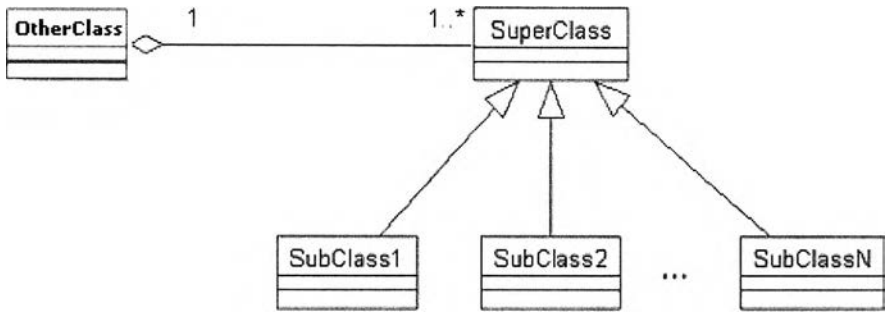
รูปที่ 3.56 ความสัมพันธ์แอริเกชันระหว่างซูเปอร์คลาสกับคลาสอื่น โดยมีซูเปอร์คลาสเป็นคลาสหลัก



รูปที่ 3.57 ความสัมพันธ์แอริเกชันระหว่างคลาสอื่นกับซูเปอร์คลาส โดยมีซูเปอร์คลาสเป็นคลาสย่อย



รูปที่ 3.58 ความสัมพันธ์คอมโพสิชันระหว่างซูเปอร์คลาสกับคลาสอื่น โดยมีซูเปอร์คลาสเป็นคลาสหลัก



รูปที่ 3.59 ความสัมพันธ์คอมโพสิชันระหว่างคลาสอื่นกับซูเปอร์คลาส มีซูเปอร์คลาสเป็นคลาสย่อย

ถ้า SuperClass มีความสัมพันธ์กับ OtherClass นั่นคือ ซับคลาสทั้งหมด ได้แก่ SubClass1, SubClass2, ..., SubClassN แต่ละซับคลาสจะมีความสัมพันธ์กับ OtherClass ด้วย เรียกความสัมพันธ์ของซับคลาสทั้งหมด โดยแต่ละซับคลาสจะมีความสัมพันธ์กับ OtherClass ว่าความสัมพันธ์โดยปริยาย นั่นคือ จะได้แอ็บสแตร็คแมชชีนของความสัมพันธ์โดยปริยายระหว่างซับคลาสกับคลาสอื่น คือ ImplicitRelTypeRelName_SubClass_OtherClass

แอ็บสแตร็คแมชชีนบี SubClass และแอ็บสแตร็คแมชชีนบี OtherClass จะระบุแอ็บสแตร็คแมชชีนบีของความสัมพันธ์โดยปริยาย ImplicitRelTypeRelName_SubClass_Other Class ในอนุประโยค USES

จากรูปที่ 3.55 ถึงรูปที่ 3.59 นำมาสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์ระหว่างซูเปอร์คลาสกับคลาสอื่น รวมทั้งแอ็บสแตร็คแมชชีนบีของความสัมพันธ์โดยปริยายของซับคลาสจากความสัมพันธ์ระหว่างซูเปอร์คลาสกับคลาสอื่น แสดงดังรูปที่ 3.60 ถึงรูปที่ 3.69

MACHINE
Asso_SuperClass_OtherClass
USES
BasicSuperClass, BasicOtherClass
VARIABLES
asso_superclass_otherclass
...
END

รูปที่ 3.60 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างซูเปอร์คลาสกับคลาสอื่น

MACHINE
ImplicitAsso_SubClass1_OtherClass
USES
BasicSubClass1, BasicOtherClass
VARIABLES
implicitasso_subclass1_otherclass
...
END

```
MACHINE
ImplicitAsso_SubClass2_OtherClass
USES
BasicSubClass2, BasicOtherClass
VARIABLES
implicitasso_subclass2_otherclass
...
END
```

...

```
MACHINE
ImplicitAsso_SubClassN_OtherClass
USES
BasicSubClassN, BasicOtherClass
VARIABLES
implicitasso_subclassN_otherclass
...
END
```

รูปที่ 3.61 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่างซับคลาสกับคลาสอื่น

```
MACHINE
Aggr_SuperClass_OtherClass
USES
BasicSuperClass, BasicOtherClass
VARIABLES
aggr_superclass_otherclass
...
END
```

รูปที่ 3.62 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอริเกชันระหว่างซูเปอร์คลาสกับคลาสอื่น

```
MACHINE
ImplicitAggr_SubClass1_OtherClass
USES
BasicSubClass1, BasicOtherClass
VARIABLES
implicitaggr_subclass1_otherclass
...
END
```

```
MACHINE
ImplicitAggr_SubClass2_OtherClass
USES
BasicSubClass2, BasicOtherClass
VARIABLES
implicitaggr_subclass2_otherclass
...
END
```

...

MACHINE
ImplicitAggr_SubClassN_OtherClass
USES
BasicSubClassN, BasicOtherClass
VARIABLES
implicitaggr_subclassN_otherclass
...
END

รูปที่ 3.63 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันโดยปริยายระหว่างซับคลาสกับคลาสอื่น

MACHINE
Aggr_OtherClass_SuperClass
USES
BasicOtherClass, BasicSuperClass
VARIABLES
aggr_otherclass_superclass
...
END

รูปที่ 3.64 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอริเกชันระหว่างคลาสอื่นกับซูเปอร์คลาส

MACHINE
ImplicitAggr_OtherClass_SubClass1
USES
BasicOtherClass, BasicSubClass1
VARIABLES
implicitaggr_otherclass_subclass1
...
END

MACHINE
ImplicitAggr_OtherClass_SubClass2
USES
BasicOtherClass, BasicSubClass2
VARIABLES
implicitaggr_otherclass_subclass2
...
END

...

MACHINE
ImplicitAggr_OtherClass_SubClassN
USES
BasicOtherClass, BasicSubClassN
VARIABLES
implicitaggr_otherclass_subclassN
...
END

รูปที่ 3.65 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันโดยปริยายระหว่างคลาสอื่นกับซับคลาส

MACHINE
Compo_SuperClass_OtherClass
USES
BasicSuperClass, BasicOtherClass
VARIABLES
compo_superclass_otherclass
...
END

รูปที่ 3.66 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างซูเปอร์คลาสกับคลาสอื่น

MACHINE
ImplicitCompo_SubClass1_OtherClass
USES
BasicSubClass1, BasicOtherClass
VARIABLES
implicitcompo_subclass1_otherclass
...
END

MACHINE
ImplicitCompo_SubClass2_OtherClass
USES
BasicSubClass2, BasicOtherClass
VARIABLES
implicitcompo_subclass2_otherclass
...
END

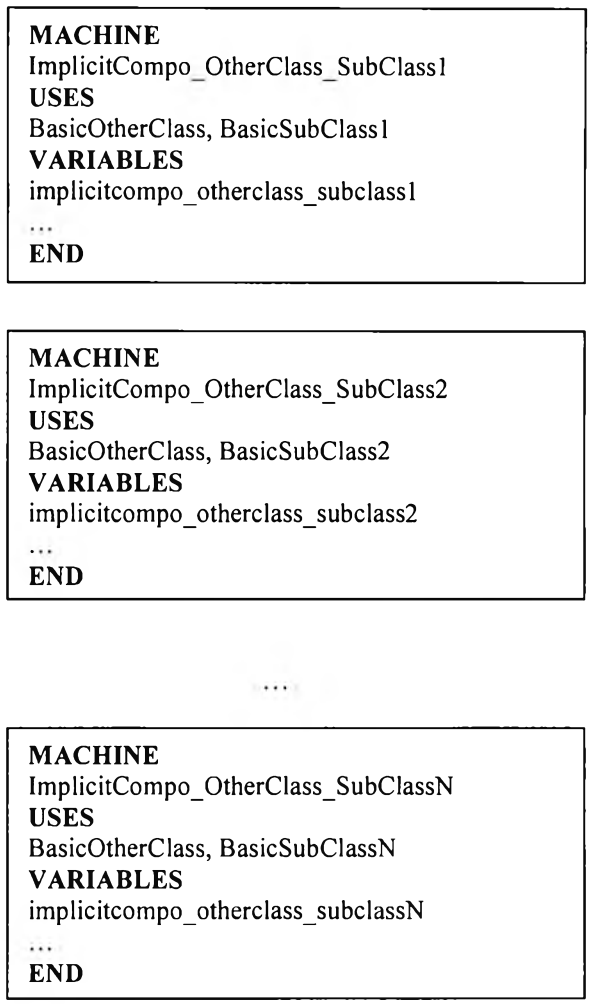
...

MACHINE
ImplicitCompo_SubClassN_OtherClass
USES
BasicSubClassN, BasicOtherClass
VARIABLES
implicitcompo_subclassN_otherclass
...
END

รูปที่ 3.67 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันโดยปริยายระหว่างซับคลาสกับคลาสอื่น

MACHINE
Compo_OtherClass_SuperClass
USES
BasicOtherClass, BasicSuperClass
VARIABLES
compo_otherclass_superclass
...
END

รูปที่ 3.68 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสอื่นกับซูเปอร์คลาส



รูปที่ 3.69 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันโดยปริยายระหว่างคลาสอื่นกับซับคลาส

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 8 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 8

ถ้า SuperClassi มีความสัมพันธ์ RelName กับ OtherClass ในแผนภาพคลาส จะทำให้ SubClassi มีความสัมพันธ์ RelName โดยปริยายกับ OtherClass เขียนแทนด้วย ImplicitRelTypeRelName_SubClassi_OtherClass โดยที่ RelType แทนความสัมพันธ์แอสโซซิเอชัน ความสัมพันธ์แอกริเกชัน และความสัมพันธ์คอมโพสิชัน

สำหรับ ImplicitRelTypeRelName_SubClassi_OtherClass สามารถพิจารณา RelType และความสัมพันธ์ระหว่าง SubClassi กับ OtherClass โดยแบ่งออกเป็น 5 กรณี ดังนี้

กรณีที่ 1 RelType = Asso

จะได้แอ็บสแตร็ตแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่าง SubClassi และ OtherClass นั่นคือ ImplicitAssoAssoName_SubClassi_OtherClass

กรณีที่ 2 RelType = Aggr **โดยมี** SubClassi **เป็นคลาสหลัก**

จะได้แอ็บสแตร็ตแมชชีนบีของความสัมพันธ์แอกริเกชันโดยปริยายระหว่าง SubClassi และ OtherClass นั่นคือ ImplicitAggrAggrName_SubClassi_OtherClass

กรณีที่ 3 RelType = Aggr **โดยมี** SubClassi **เป็นคลาสย่อย**

จะได้แอ็บสแตร็ตแมชชีนบีของความสัมพันธ์แอกริเกชันโดยปริยายระหว่าง OtherClass และ SubClassi นั่นคือ ImplicitAggrAggrName_OtherClass_SubClassi

กรณีที่ 4 RelType = Compo **โดยมี** SubClassi **เป็นคลาสหลัก**

จะได้แอ็บสแตร็ตแมชชีนบีของความสัมพันธ์คอมโพสิชันโดยปริยายระหว่าง SubClassi และ OtherClass นั่นคือ ImplicitCompoCompoName_SubClassi_OtherClass

กรณีที่ 5 RelType = Compo **โดยมี** SubClassi **เป็นคลาสย่อย**

จะได้แอ็บสแตร็ตแมชชีนบีของความสัมพันธ์คอมโพสิชันโดยปริยายระหว่าง OtherClass และ SubClassi นั่นคือ ImplicitCompoCompoName_OtherClass_SubClassi

จากข้อความข้างต้น จะสามารถทำการสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์โดยปริยายระหว่าง SubClassi กับ OtherClass คือ ImplicitRelTypeRelName_SubClassi_OtherClass ได้ดังนี้

R8-1) กำหนดอนุประโยค MACHINE

R8-2) กำหนดชื่อของแอ็บสแตร็คแมชชีนบีของความสัมพันธ์โดยปริยายระหว่าง SubClassi และ OtherClass คือ ImplicitRelTypeRelName_SubClassi_OtherClass ในอนุประโยค MACHINE

R8-3) กำหนดอนุประโยค USES

R8-4) ระบุชื่อของแอ็บสแตร็คแมชชีนบี BasicSubClassi และ BasicOtherClass ในอนุประโยค USES

R8-5) จากข้อที่ R8-1 กำหนดเซตของตัวแปรของความสัมพันธ์ใด ๆ โดยปริยายระหว่าง SubClassi และ OtherClass คือ implicitreltyperelname_subclassi_otherclass ในอนุประโยค VARIABLES

R8-6) กำหนดความสัมพันธ์ระหว่างเซตของตัวแปรของความสัมพันธ์ใด ๆ โดยปริยายระหว่าง SubClassi และ OtherClass คือ implicitreltyperelname_subclassi_otherclass และผลคูณคาร์ทีเซียนระหว่างเซตของแอ็บสแตร็คแมชชีน BasicSubClassi และ BasicOtherClass นั่นคือ BASICSUBCLASSi × BASICOTHERCLASS

จะได้ว่า

implicitreltyperelname_subclassi_otherclass ⊂

BASICSUBCLASSi × BASICOTHERCLASS .....(1)

กำหนด dom(implicitreltyperelname_subclassi_otherclass) = basicsubclassi.....(2)

และ ran(implicitreltyperelname_subclassi_otherclass) = basicotherclass.....(3)

กำหนดตัวแปร xx และ yy ภายใต้ตัวบ่งปริมาณ"ทั้งหมด" นั่นคือ ∀ (xx,yy)

ทำการสร้างเงื่อนไขของตัวแปร xx และ yy คือ

xx ∈ dom(implicitreltyperelname_subclassi_otherclass) ∧

yy ∈ ran(implicitreltyperelname_subclassi_otherclass) ภายใต้ตัวบ่งปริมาณทั้งหมดของตัวแปร xx และ yy นั่นคือ

∀ (xx,yy) (xx ∈ dom(implicitreltyperelname_subclassi_otherclass) ∧

yy ∈ ran(implicitreltyperelname_subclassi_otherclass))

ทำการกำหนดจำนวนเซตของตัวแปรของความสัมพันธ์ใด ๆ โดยปริยายระหว่าง SubClassi และ OtherClass หรือ implicitreltyperelname_subclassi_otherclass ของตัวแปร xx และ yy เพื่อระบุมัลติพลิซิตีของ SubClassi และ OtherClass นั่นคือ

card((implicitreltyperelname_subclassi_otherclass)[{xx}])

แทนมัลติพลิซิตีของ OtherClass และ

card((implicitreltyperelname_subclassi_otherclass)$^{-1}$[{yy}])

แทนมัลติพลิซิตีของ SubClassi

โดยมัลติพลิซิตีของ SubClassi และ OtherClass จะเป็นผลจากการระบุเงื่อนไขของตัวแปร xx และ yy ภายใต้ภายใต้ตัวบ่งปริมาณทั้งหมดของตัวแปร xx และ yy นั่นคือ

∀ (xx,yy) (((xx ∈ dom(implicitreltyperelname_subclassi_otherclass) ∧

(yy ∈ ran(implicitreltyperelname_subclassi_otherclass))

→ card((implicitreltyperelname_subclassi_otherclass)[{xx}]) =

Multiplicity of OtherClass ∧

card((implicitreltyperelname_subclassi_otherclass)$^{-1}$[{yy}]) =

Multiplicity of SubClassi) .....(4)

นำ (1) ∧ (2) ∧ (3) ∧ (4) จะได้ว่า

implicitreltyperelname_subclassi_otherclass ⊂

BASICSUBCLASSi × BASICOTHERCLASS ∧

dom(implicitreltyperelname_subclassi_otherclass) = basicsubclassi ∧

ran(implicitreltyperelname_subclassi_otherclass) = basicotherclass ∧

$\forall$ (xx,yy) (((xx $\in$ dom(implicitreltyperelname_subclassi_otherclass) $\wedge$

(yy $\in$ ran(implicitreltyperelname_subclassi_otherclass))

$\rightarrow$ card((implicitreltyperelname_subclassi_otherclass)[{xx}]) =

Multiplicity of OtherClass $\wedge$

card((implicitreltyperelname_subclassi_otherclass)$^{-1}$[{yy}]) =

Multiplicity of SubClassi)

ในอนุประโยค INVARIANT

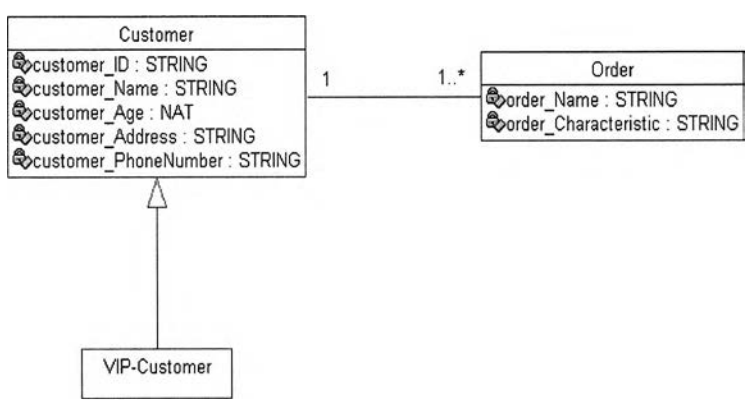
R8-7) กำหนดอนุประโยค INITIALISATION

R8-8) กำหนดค่าเริ่มต้นของเซตของตัวแปรของความสัมพันธ์ใด ๆ โดยปริยายระหว่าง SubClassi และ OtherClass คือ implicitreltyperelname_subclassi_otherclass := ϕ ในอนุประโยค INITIALISATION

R8-9) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี ImplicitRelTypeRelName_SubClassi_OtherClass

R8-10) ระบุแอ็บสแตร็คแมชชีนบี ImplicitRelTypeRelName_SubClassi_OtherClass ในอนุประโยค USES ของแอ็บสแตร็คแมชชีนบี SubClassi และ OtherClass

จากตัวอย่างคลาส Customer ซึ่งเป็นซูเปอร์คลาส โดยถ่ายทอดคุณลักษณะทั้งหมดให้กับคลาส VIP-Customer ซึ่งเป็นซับคลาส โดยคลาส Customer มีความสัมพันธ์แบบแอสโซซิเอชัน กับคลาส Order แสดงได้ดังรูปที่ 3.70



รูปที่ 3.70 ตัวอย่างคลาส Customer เป็นซูเปอร์คลาสที่มีความสัมพันธ์แอสโซซิเอชันกับคลาส Order

จากรูปที่ 3.70 แสดงตัวอย่างคลาส Customer ซึ่งถ่ายทอดคุณลักษณะทั้งหมดให้กับคลาส VIP-Customer โดยที่คลาส Customer มีความสัมพันธ์แอสโซซิเอชันกับคลาส Order

จากคลาส Customer ซึ่งเป็นซูเปอร์คลาสจะมีความสัมพันธ์แอสโซซิเอชันกับคลาส Order จะทำให้คลาส VIP – Customer ซึ่งเป็นซับคลาสย่อมมีความสัมพันธ์แอสโซซิเอชันโดยปริยายกับคลาส Order สามารถนำมาสร้างแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาส Customer กับคลาส Order และแอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายของคลาส VIP-Customer กับคลาส Order แสดงได้ดังรูปที่ 3.71 และรูปที่ 3.72

MACHINE
Asso_Customer_Order
USES
BasicCustomer, BasicOrder
VARIABLES
asso_customer_order
INVARIANT
asso_customer_order $\subset$ BASICCUSTOMER $\times$ BASICORDER $\wedge$
dom(asso_customer_order) = basiccustomer $\wedge$
ran(asso_customer_order) = basicorder $\wedge$
$\forall$ (xx,yy).(((xx $\in$ dom(asso_customer_order)) $\wedge$
(yy $\in$ ran(asso_customer_order)))
$\rightarrow$ card((asso_customer_order)[{xx}]) $\geq$ 1 $\wedge$
card((asso_customer_order)$^{-1}$[{yy}]) = 1)
INITIALISATION
asso_customer_order := ϕ
END

รูปที่ 3.71 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาส Customer กับคลาส Order

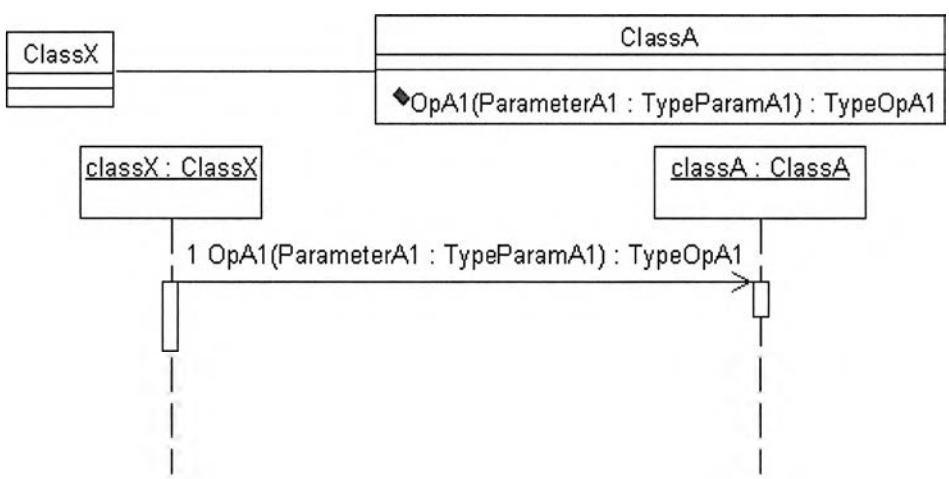
MACHINE
ImplicitAsso_VIP-Customer_Order
USES
BasicVIP-Customer, BasicOrder
VARIABLES
implicitasso_vip-customer_order
INVARIANT
implicitasso_vip-customer_order $\subset$ BASICVIP-CUSTOMER $\times$ BASICORDER $\wedge$
dom(implicitasso_vip-customer_order) = basicvip-customer $\wedge$
ran(implicitasso_vip-customer_order) = basicorder $\wedge$
$\forall$ (xx,yy).(((xx $\in$ dom(implicitasso_vip-customer_order)) $\wedge$
(yy $\in$ ran(implicitasso_vip-customer_order)))
$\rightarrow$ card((implicitasso_vip-customer_order)[{xx}]) $\geq$ 1 $\wedge$
card((implicitasso_vip-customer_order)$^{-1}$[{yy}]) = 1)
INITIALISATION
implicitasso_vip-customer_order := ϕ
END

รูปที่ 3.72 แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่างคลาส VIP-Customer กับคลาส Order

3.3 โอเปอเรชันของคลาสที่แสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คจากเหตุการณ์ในระบบของแผนภาพซีเควนซ์กับกฎการแปลง

จากรูปแบบโอเปอเรชันของคลาสที่แสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คจากเหตุการณ์ในระบบของแผนภาพซีเควนซ์ แบ่งหัวข้อการนำเสนอเป็น 3 ส่วน ได้แก่ การระบุโอเปอเรชันของคลาสในแอ็บสแตร็คแมชชีนบี โดยพิจารณาจากรูปแบบการรับส่งข้อความระหว่างออบเจ็คจากเหตุการณ์ของแผนภาพซีเควนซ์ และการสร้างแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสเพื่อทำการระบุโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชันภายในคลาสเดิม จะอธิบายในหัวข้อที่ 3.3.1 โดยประกอบด้วย กฎข้อที่ 9 และกฎข้อที่ 10 การถ่ายทอดโอเปอเรชันจากซูเปอร์คลาส จะอธิบายในหัวข้อที่ 3.3.2 ประกอบด้วย กฎข้อที่ 11 และการสร้างอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี รวมทั้งการสร้างอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีซับคลาส เพื่อแสดงการเรียกโอเปอเรชันของคลาสที่แสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คจากเหตุการณ์ต่างๆ ของแผนภาพซีเควนซ์ รวมทั้งการระบุเงื่อนไขที่เป็นจริงของโอเปอเรชันจากเหตุการณ์ของแผนภาพซีเควนซ์จะอธิบายในหัวข้อที่ 3.3.3 ประกอบด้วย กฎข้อที่ 12 กฎข้อที่ 13 และกฎข้อที่ 14

พิจารณาโอเปอเรชันในแผนภาพคลาสโดยแสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คจากเหตุการณ์ของแผนภาพซีเควนซ์ แสดงได้ดังรูปที่ 3.73



รูปที่ 3.73 แผนภาพคลาสและแผนภาพซีเควนซ์แสดงการรับส่ง OpA() ระหว่าง classX กับ classA

จากรูปที่ 3.73 จะแสดงแนวคิดของแอ็บสแตร็คแมชชีนบีคลาสที่มีโอเปอเรชันของคลาส โดยพิจารณาจากรูปแบบการรับส่งข้อความระหว่างออบเจ็คจากเหตุการณ์ของแผนภาพซีเควนซ์ คือ แอ็บสแตร็คแมชชีนบี BasicClassA สามารถระบุโอเปอเรชัน OpA1(Parameter1 : TypeParam1) : TypeOp1 ในอนุประโยค OPERATIONS

3.3.1 การระบุโอเปอเรชันของคลาสในแอ็บสแตร็คแมชชีนบี และการสร้างแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสเพื่อระบุโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชันภายในคลาสเดิม

ในการระบุโอเปอเรชันของคลาสในแอ็บสแตร็คแมชชีนบี จะพิจารณาจากรูปแบบการรับส่งข้อความระหว่างออบเจ็คจากเหตุการณ์ต่าง ๆ ของแผนภาพซีเควนซ์ และสามารถทำการสร้างแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาส เพื่อระบุโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชันภายในคลาสเดิม โดยนำเสนอในกฎข้อที่ 9 และกฎข้อที่ 10 นั่นคือ

กฎข้อที่ 9

การระบุโอเปอเรชันเบซิคในแอ็บสแตร็คแมชชีนบีเบซิคคลาสและการระบุโอเปอเรชันนอนเบซิคในแอ็บสแตร็คแมชชีนบีคลาส

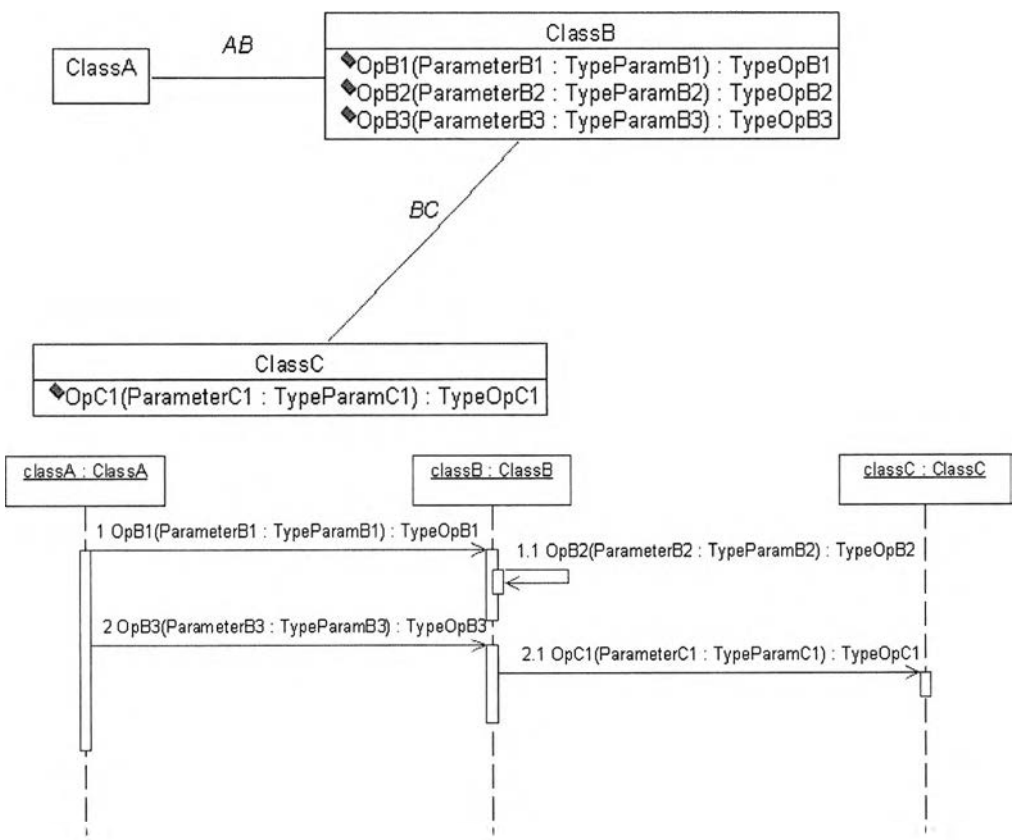
โอเปอเรชันเบซิคเป็นโอเปอเรชันที่ไม่มีการเรียกโอเปอเรชันอื่น ๆ ของคลาสได้อีก ทำการระบุโอเปอเรชันเบซิคเหล่านี้ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบีเบซิคคลาส โดยใส่คำนำหน้า คือ Basic คั่นด้วยเครื่องหมายอันเดอร์สกอร์ (Under Score) แล้วตามด้วยชื่อของโอเปอเรชันเบซิคในแอ็บสแตร็คแมชชีนบีเบซิคคลาส ในกรณีที่โอเปอเรชันจากแผนภาพคลาสและแผนภาพซีเควนซ์มีชนิดค่าที่ส่งคืน (Return Type) เป็นสายอักขระหรือจำนวนเต็มที่มีค่าตั้งแต่ศูนย์ จะระบุตัวแปรผลลัพธ์ (Output Variable) จะใส่คำนำหน้า คือ basic แล้วตามด้วยชื่อคลาสโดยใช้ตัวพิมพ์เล็ก และระบุลำดับของโอเปอเรชัน สำหรับโอเปอเรชันที่มีชนิดค่าที่ส่งคืนเป็นบูลีน จะระบุตัวตัวแปรผลลัพธ์เป็น boolean เท่านั้น ส่วนในกรณีที่โอเปอเรชันจากแผนภาพคลาสและแผนภาพซีเควนซ์ไม่มีชนิดค่าที่ส่งคืน จะระบุตัวแปรผลลัพธ์ โดยใส่คำนำหน้า คือ voidbasic แล้วตามด้วยชื่อคลาสโดยใช้ตัวพิมพ์เล็ก และระบุลำดับของโอเปอเรชันเช่นเดียวกัน

ส่วนโอเปอเรชันนอนเบซิคเป็นโอเปอเรชันที่มีการเรียกโอเปอเรชันอื่น ๆ ของคลาส ทำการระบุโอเปอเรชันนอนเบซิคเหล่านี้ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบีคลาส สำหรับโอเปอเรชันที่มีชนิดค่าที่ส่งคืนเป็นสายอักขระและจำนวนเต็มที่มีค่าตั้งแต่ศูนย์ ตัวแปรผลลัพธ์จะไม่ใส่คำนำหน้า แต่จะมีชื่อคลาสซึ่งใช้ตัวพิมพ์เล็กเท่านั้น และระบุลำดับของโอเปอเรชัน สำหรับโอเปอเรชันที่มีชนิดค่าที่ส่งคืนเป็นบูลีน จะระบุตัวตัวแปรผลลัพธ์เป็น boolean เท่านั้น ส่วนในกรณีที่โอเปอเรชันจากแผนภาพคลาสและแผนภาพซีเควนซ์ไม่มีชนิดค่าที่ส่งคืน จะระบุตัวแปรผลลัพธ์ โดยใส่คำนำหน้า คือ void แล้วตามด้วยชื่อคลาสโดยใช้ตัวพิมพ์เล็ก และระบุลำดับของโอเปอเรชันเช่นเดียวกัน

โอเปอเรชันเบซิคและโอเปอเรชันนอนเบซิคถ้ามีพารามิเตอร์และชนิดของพารามิเตอร์ในอนุประโยค OPERATIONS จะต้องระบุเงื่อนไขก่อนกระทำ (Pre-Condition) ที่แสดงถึงความสัมพันธ์ระหว่างพารามิเตอร์และชนิดของพารามิเตอร์ของโอเปอเรชัน ในส่วนของอนุประโยค PRE..THEN ที่อยู่ในอนุประโยค OPERATIONS แต่ถ้าไม่มีพารามิเตอร์และชนิดของพารามิเตอร์ของโอเปอเรชัน

ไม่ต้องระบุเงื่อนไขก่อนกระทำ ในส่วนของอนุประโยค BEGIN...END ที่อยู่ในอนุประโยค OPERATIONS

พิจารณาแผนภาพคลาสที่แสดงความสัมพันธ์ระหว่าง ClassA กับ ClassB และความสัมพันธ์ระหว่าง ClassB กับ ClassC รวมทั้งแผนภาพซีเควนซ์ที่แสดงเหตุการณ์ของการรับส่งข้อความระหว่างออบเจ็ค classA ของคลาส ClassA กับออบเจ็ค classB ของคลาส ClassB และการรับส่งข้อความกันระหว่างออบเจ็ค classB ของคลาส ClassB กับออบเจ็ค classC ของคลาส ClassC แสดงได้ดังรูปที่ 3.74



รูปที่ 3.74 รูปแบบแผนภาพคลาสและแผนภาพซีเควนซ์ที่อธิบายในกฎข้อที่ 9

จากรูปที่ 3.74 จะพบว่าในแผนภาพคลาสจะประกอบไปด้วย ClassA ClassB และ ClassC โดยที่ ClassB มีโอเปอเรชัน OpB1(ParameterB1 : TypeParamB1) : TypeOpB1 โอเปอเรชัน OpB2(ParameterB2 : TypeParamB2) : TypeOpB2 และ โอเปอเรชัน OpB3(ParameterB3 : TypeParamB3) : TypeOpB3 และ ClassC จะประกอบไปด้วยโอเปอเรชัน OpC1(ParameterC1 : TypeParamC1) : TypeOpC1 จากแผนภาพซีเควนซ์ที่แสดงเหตุการณ์ของการรับส่งข้อความระหว่างออบเจ็ค classA ของ ClassA กับ ออบเจ็ค classB ของ ClassB และการรับส่งข้อความระหว่างออบเจ็ค classB ของ ClassB กับ ออบเจ็ค classC ของ ClassC จะพบว่า

โอเปอเรชัน OpB2( ) ของ ClassB เป็นโอเปอเรชันเบซิค และโอเปอเรชัน OpC1( ) ของ ClassC เป็นโอเปอเรชันเบซิค สามารถทำการระบุโอเปอเรชันเบซิคเหล่านี้ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี BasicClassB และแอ็บสแตร็คแมชชีนบี BasicClassC ตามลำดับ โดยในอนุประโยค OPERATIONS จะมีตัวแปรผลลัพธ์ ซึ่งแสดงชนิดค่าที่ส่งคืนของโอเปอเรชัน หรือ TypeOpB2 คือ basicclassb1 และชนิดค่าที่ส่งคืนของโอเปอเรชันหรือ TypeOpC1 คือ basicclassc1 แสดงได้ดังรูปที่ 3.75 และรูปที่ 3.76

ส่วนโอเปอเรชัน OpB1( ) และโอเปอเรชัน OpB3( ) เป็นโอเปอเรชันนอนเบซิค โดยทำการระบุโอเปอเรชันนอนเบซิคเหล่านี้ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี ClassB โดยตัวแปรผลลัพธ์ ซึ่งแสดงชนิดค่าที่ส่งคืนของโอเปอเรชันหรือ TypeOpB1 คือ classb1 และแสดงชนิดค่าที่ส่งคืนของโอเปอเรชันหรือ TypeOpB3 คือ classb2 แสดงได้ดังรูปที่ 3.77

```
MACHINE
BasicClassB
...
OPERATIONS
basicclassb1 ← Basic_OpB2(ParameterB2) =
PRE
ParameterB2 ∈ TypeParamB2
THEN
...
END

END
```

รูปที่ 3.75 ระบุโอเปอเรชัน Basic_OpB2() ในแอ็บสแตร็คแมชชีนบี BasicClassB

```
MACHINE
BasicClassC
...
OPERATIONS
basicclassc1 ← Basic_OpC1(ParameterC1) =
PRE
ParameterC1 ∈ TypeParamC1
THEN
...
END

END
```

รูปที่ 3.76 ระบุโอเปอเรชัน Basic_OpB1() ในแอ็บสแตร็คแมชชีนบี BasicClassC

```
MACHINE
ClassB
...
OPERATIONS
classb1 ← OpB1(ParameterB1) =
PRE
ParameterB1 ∈ TypeParamB1
THEN
...
END;

classb2 ← OpB3(ParameterB3) =
PRE
ParameterB3 ∈ TypeParamB3
THEN
...
END

END
```

รูปที่ 3.77 ระบุโอเปอเรชัน OpB1() และ OpB3() ในแอ็บสแตร็คแมชชีนบี ClassB

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 9 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 9

สำหรับ Classi Classj และ ClassX ในแผนภาพคลาส โดยที่ Classi จะมีโอเปอเรชัน Opi() : TypeOpi และ Classj จะมีโอเปอเรชัน Opj() : TypeOpj พิจารณาในแผนภาพคลาสจะพบว่า ClassX จะมีความสัมพันธ์กับ Classi และ Classi จะมีความสัมพันธ์กับ Classj จะสามารถแสดงได้ในเหตุการณ์ที่ k ของแผนภาพซีเควนซ์ นั่นคือ ออบเจ็ค classx จะมีการร้องขอให้ออบเจ็ค classi ของคลาส Classi ทำโอเปอเรชัน Opi() และออบเจ็ค classi ของคลาส Classi จะมีการร้องขอให้ออบเจ็ค classj ของคลาส Classj ทำโอเปอเรชัน Opj() โดยที่โอเปอเรชัน Opi() จะเป็นโอเปอเรชันที่เรียกโอเปอเรชัน Opj()

จากข้อความข้างต้นจะสามารถทำการระบุโอเปอเรชัน Opi() และโอเปอเรชัน Opj() จากแผนภาพคลาสที่แสดงในเหตุการณ์ที่ k ของแผนภาพซีเควนซ์ในแอ็บสแตร็คแมชชีนบี ดังนี้

Opi() เป็นโอเปอเรชันนอนเบซิค

R9-1) พิจารณาแอ็บสแตร็คแมชชีนบี Classi

R9-2) กำหนดอนุประโยค OPERATIONS

R9-3) ระบุโอเปอเรชัน Opi() ในอนุประโยค OPERATIONS โดยพิจารณาข้อที่ R9-4 และ R9-5

R9-4) พิจารณาโอเปอเรชัน Opi() มีพารามิเตอร์ของโอเปอเรชัน Opi() หรือไม่ ดังนี้

R9-4-1) ถ้าโอเปอเรชัน Opi() มีพารามิเตอร์ ให้กำหนดอนุประโยค PRE โดยทำการระบุ

เงื่อนไขของความสัมพันธ์ระหว่างพารามิเตอร์และชนิดของพารามิเตอร์ นั่นคือ Parameteri ∈ TypeParami แล้วกำหนดอนุประโยค THEN

R9-4-2) ถ้าโอเปอเรชัน Opi() ไม่มีพารามิเตอร์ ให้กำหนดอนุประโยค BEGIN

R9-5) พิจารณาชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi() ดังนี้

R9-5-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi() เป็น STRING หรือ NAT ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ classi1

R9-5-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi() เป็น BOOLEAN ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ boolean

R9-5-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi() เป็น void ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ voidclassi1

R9-6) จากข้อที่ R9-5 ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Opi() ในแอ็บสแตร็คแมชชีนบี Classi ดังนี้

R9-5-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi() เป็น STRING หรือ void ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Opi() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ classi1 := null หรือ voidclassi1 := null ตามลำดับ

R9-5-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi() เป็น BOOLEAN ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Opi() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ boolean := FALSE

R9-5-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi() เป็น NAT ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Opi() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ classi1 := 0

R9-6) กำหนดอนุประโยค END เพื่อจบการทำงานของอนุประโยค OPERATIONS

Opj() เป็นโอเปอเรชันเบซิค

R9-7) พิจารณาแอ็บสแตร็คแมชชีนบี BasicClassj

R9-8) กำหนดอนุประโยค OPERATIONS

R9-9) ระบุโอเปอเรชัน Basic_Opj() ในอนุประโยค OPERATIONS โดยพิจารณาข้อที่ R9-10 และ R9-11

R9-10) พิจารณาโอเปอเรชัน Basic_Opj() มีพารามิเตอร์ของโอเปอเรชัน Basic_Opi() หรือไม่ ดังนี้

R9-10-1) ถ้าโอเปอเรชัน Basic_Opj() มีพารามิเตอร์ ให้กำหนดอนุประโยค PRE โดยทำการระบุเงื่อนไขของความสัมพันธ์ระหว่างพารามิเตอร์และชนิดของพารามิเตอร์ นั่นคือ Parameterj ∈ TypeParamj แล้วกำหนดอนุประโยค THEN

R9-10-2) ถ้าโอเปอเรชัน Basic_Opj() ไม่มีพารามิเตอร์ ให้กำหนดอนุประโยค BEGIN

R9-11) พิจารณาชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() ดังนี้

R9-11-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi() เป็น STRING หรือ NAT ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicClassj คือ basicclassj1

R9-11-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() เป็น BOOLEAN ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicClassj คือ boolean

R9-11-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() เป็น void ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicClassj คือ voidbasicclassj1

R9-12) จากข้อที่ R9-11 ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Basic_Opj() ในแอ็บสแตร็คแมชชีนบี BasicClassj ดังนี้

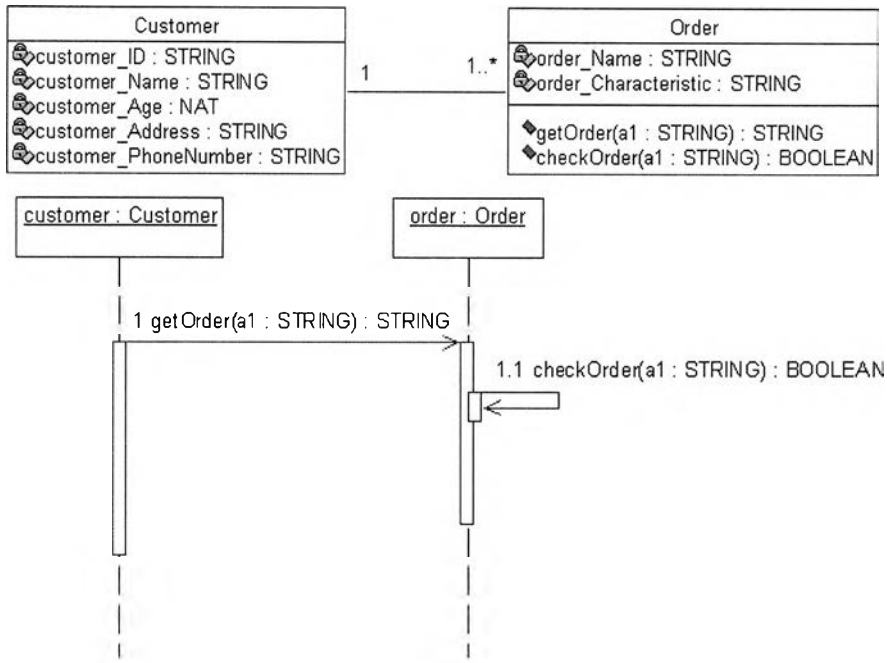
R9-12-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() เป็น STRING หรือ void ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Basic_Opj() ในอนุประโยค THENของแอ็บสแตร็คแมชชีนบี BasicClassj คือ basicclassj1 := null หรือ voidbasicclassj1 := null ตามลำดับ

R9-12-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() เป็น BOOLEAN ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Basic_Opj() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicClassj คือ boolean := FALSE

R9-12-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() เป็น NAT ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Basic_Opj() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicClassj คือ basicclassj1 := 0

R9-13) กำหนดอนุประโยค END เพื่อจบการทำงานของอนุประโยค OPERATIONS

จากตัวอย่างของแผนภาพคลาสที่แสดงส่วนหนึ่งของระบบการสั่งซื้อสินค้า ซึ่งประกอบไปด้วยคลาส Customer และคลาส Order รวมทั้งแผนภาพซีเควนซ์แสดงเหตุการณ์การสร้างรายการของการสั่งซื้อสินค้าจากลูกค้า แสดงได้ดังรูปที่ 3.78



รูปที่ 3.78 ตัวอย่างแผนภาพคลาสและแผนภาพซีเควนซ์ของเหตุการณ์การสั่งซื้อสินค้าที่อธิบายในกฎข้อที่ 9

จากรูปที่ 3.78 จะพบว่า คลาส Order จะประกอบไปด้วยโอเปอเรชัน getOrder(a1 : STRING) : STRING และโอเปอเรชัน checkOrder(a1 : STRING) : BOOLEAN และจากแผนภาพซีเควนซ์ที่แสดงเหตุการณ์การสร้างรายการของการสั่งซื้อสินค้าจากลูกค้า จะพบว่ามีการรับส่งข้อความระหว่างออบเจ็ค customer ของคลาส Customer กับออบเจ็ค order ของคลาส Order เมื่อพิจารณาโอเปอเรชันของคลาส Order จะพบว่าโอเปอเรชัน checkOrder( ) เป็นโอเปอเรชันเบซิค ระบุโอเปอเรชันเบซิค checkOrder( ) ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี BasicOrder แสดงได้ดังรูปที่ 3.79

และโอเปอเรชัน getOrder( ) เป็นโอเปอเรชันนอนเบซิค ระบุโอเปอเรชันนอนเบซิค getOrder ( ) ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี Order แสดงได้ดังรูปที่ 3.80

```
MACHINE
BasicOrder
...
OPERATIONS
boolean ← Basic_checkOrder(a1) =
PRE
a1 ∈ STRING
THEN
...
END

END
```

รูปที่ 3.79 ระบุโอเปอเรชัน Basic_checkOrder( ) ในแอ็บสแตร็คแมชชีนบี BasicOrder

```
MACHINE
Order
...
OPERATIONS
order1 ← getOrder(a1) =
PRE
a1 ∈ STRING
THEN
...
END

END
```

รูปที่ 3.80 ระบุโอเปอเรชัน getOrder( ) ในแอ็บสแตร็คแมชชีนบี Order

กฎข้อที่ 10

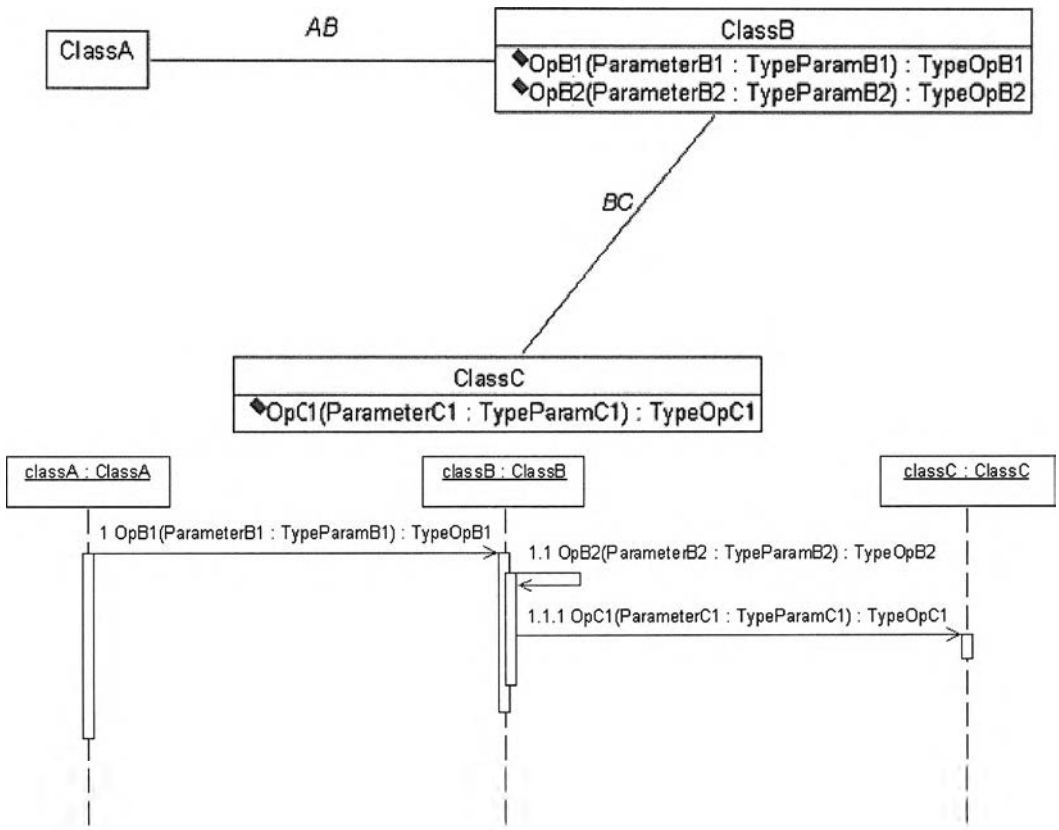
การสร้างแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสเพื่อระบุโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชันภายในคลาสเดิม

ในกรณีที่มีโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชันภายในคลาสเดิม จะสามารถทำการสร้างแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาส โดยที่แอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสเป็นแอ็บสแตร็คแมชชีนบีที่มีลักษณะเหมือนกับแอ็บสแตร็คแมชชีนบีคลาส

สำหรับโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชันภายในคลาสเดิม จะระบุโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชันภายภายในคลาสเดิมนี้ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาส โดยมีรูปแบบคือใส่คำนำหน้า Intermediate คั่นด้วยเครื่องหมายอันเดอร์สกอร์ แล้วตามด้วยชื่อของโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชันภายในคลาสเดิมในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาส

ส่วนการระบุตัวแปรผลลัพธ์ โดยการพิจารณาจากชนิดค่าที่ส่งคืนของโอเปอเรชันจะทำได้เช่นเดียวกับการระบุตัวแปรผลลัพธ์ของโอเปอเรชันในแอ็บสแตร็คแมชชีนบีคลาส โดยตัวแปรผลลัพธ์จะใส่คำนำหน้า คือ intermediate ตามด้วยชื่อคลาสโดยใช้ตัวพิมพ์เล็ก แล้วตามด้วยลำดับของโอเปอเรชัน

พิจารณาจากแผนภาพคลาสที่แสดงความสัมพันธ์ระหว่าง ClassA กับ ClassB และความสัมพันธ์ระหว่าง ClassB กับ ClassC รวมทั้งแผนภาพซีเควนซ์ที่แสดงเหตุการณ์ของการรับส่งข้อความระหว่างออบเจ็ค classA ของคลาส ClassA กับออบเจ็ค classB ของคลาส ClassB และการรับส่งข้อความระหว่างออบเจ็ค classB ของคลาส ClassB กับออบเจ็ค classC ของคลาส ClassC แสดงได้ดังรูปที่ 3.81



รูปที่ 3.81 รูปแบบแผนภาพคลาสและแผนภาพซีเควนซ์ที่อธิบายในกฎข้อที่ 10

จากรูปที่ 3.81 จะพบว่าในแผนภาพคลาสจะประกอบไปด้วย ClassA ClassB และ ClassC โดยที่ ClassB มีโอเปอเรชัน OpB1(ParameterB1 : TypeParamB1) : TypeOpB1 และโอเปอเรชัน OpB2(ParameterB2 : TypeParamB2) : TypeOpB2 ส่วน ClassC มีโอเปอเรชัน OpC1(Parameter C1 : TypeParamC1) : TypeOpC1 จากแผนภาพซีเควนซ์ที่แสดงเหตุการณ์ของการรับส่งข้อความระหว่างออบเจ็ค จะพบว่าโอเปอเรชัน OpC1( ) ของ ClassC เป็นโอเปอเรชันเบซิค ระบุโอเปอเรชัน Basic_OpC1() ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี BasicClassC แสดงได้ดังรูปที่ 3.82

ส่วนโอเปอเรชัน OpB1( ) และโอเปอเรชัน OpB2( ) เป็นโอเปอเรชันนอนเบซิค ซึ่งโอเปอเรชัน OpB1( ) จะถูกระบุในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี ClassB ส่วนโอเปอเรชัน OpB2( ) จะเป็นโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชัน OpB1( ) ภายในคลาสเดิม ทำการสร้างแอ็บสแตร็คแมชชีนบี IntermediateClassB และระบุโอเปอเรชัน Intermediate_OpB2( ) นี้ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี IntermediateClassB แสดงได้ดังรูปที่ 3.83 และรูปที่ 3.84

```
MACHINE
BasicClassC
...
OPERATIONS
basicclassc1 ← Basic_OpC1(ParameterC1) =
PRE
ParameterC1 ∈ TypeParamC1
THEN
...
END

END
```

รูปที่ 3.82 ระบุโอเปอเรชัน Basic_OpC1() ในแอ็บสแตร็คแมชชีนบี BasicClassC

```
MACHINE
ClassB
...
OPERATIONS
classb1 ← OpB1(ParameterB1) =
PRE
ParameterB1 ∈ TypeParamB1
THEN
...
END

END
```

รูปที่ 3.83 ระบุโอเปอเรชัน OpB1() ในแอ็บสแตร็คแมชชีนบี ClassB

```
MACHINE
IntermeidateClassB
...
OPERATIONS
intermeidateclassb1 ← Intermediate_OpB2(ParameterB2) =
PRE
ParameterB2 ∈ TypeParamB2
THEN
...
END

END
```

รูปที่ 3.84 สร้างแอ็บสแตร็คแมชชีนบี IntermediateClassB ทำการระบุโอเปอเรชัน Intermediate_OpB2()

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 10 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 10

สำหรับ Classi Classj และ ClassX ในแผนภาพคลาส โดยที่ Classi จะมีโอเปอเรชัน Opi1() : TypeOpi1 และ Opi2() : TypeOpi2 ส่วน Classj จะมีโอเปอเรชัน Opj() : TypeOpj พิจารณาใน

แผนภาพคลาสจะพบว่า ClassX จะมีความสัมพันธ์กับ Classi และ Classi จะมีความสัมพันธ์กับ Classj จะสามารถแสดงได้ในเหตุการณ์ที่ k ของแผนภาพซีเควนซ์ นั่นคือ ออบเจ็ค classx ของคลาส ClassX จะมีการร้องขอให้ออบเจ็ค classi ของคลาส Classi ทำโอเปอเรชัน Opi1() ซึ่งออบเจ็ค classi จะทำโอเปอเรชัน Opi2() และออบเจ็ค classi จะมีการร้องขอให้ออบเจ็ค classj ของคลาส Classj ทำโอเปอเรชัน Opj() โดยที่ โอเปอเรชัน Opi1() จะเป็นโอเปอเรชันที่เรียกโอเปอเรชัน Opi2() และโอเปอเรชัน Opi2() จะเป็นโอเปอเรชันที่เรียกโอเปอเรชัน Opj()

จากข้อความข้างต้นจะสามารถทำการระบุโอเปอเรชัน Opi1() โอเปอเรชัน Opi2() และโอเปอเรชัน Opj() จากแผนภาพคลาสที่แสดงในเหตุการณ์ที่ k ของแผนภาพซีเควนซ์ในแอ็บสแตร็คแมชชีนบี ดังนี้

Opi1() เป็นโอเปอเรชันนอนเบซิค

R10-1) พิจารณาแอ็บสแตร็คแมชชีนบี Classi

R10-2) กำหนดอนุประโยค OPERATIONS

R10-3) ระบุโอเปอเรชัน Opi1() ในอนุประโยค OPERATIONS โดยพิจารณาข้อที่ R10-4 และ R10-5

R10-4) พิจารณาโอเปอเรชัน Opi1() มีพารามิเตอร์ของโอเปอเรชัน Opi1() หรือไม่ ดังนี้

R10-4-1) ถ้าโอเปอเรชัน Opi1() มีพารามิเตอร์ ให้กำหนดอนุประโยค PRE โดยทำการระบุเงื่อนไขของความสัมพันธ์ระหว่างพารามิเตอร์และชนิดของพารามิเตอร์ นั่นคือ Parameteri1 ∈ TypeParami1 แล้วกำหนดอนุประโยค THEN

R10-4-2) ถ้าโอเปอเรชัน Opi1() ไม่มีพารามิเตอร์ ให้กำหนดอนุประโยค BEGIN

R10-5) พิจารณาชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi1() ดังนี้

R10-5-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi1() เป็น STRING หรือ NAT ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ classi1

R10-5-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi1() เป็น BOOLEAN ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ boolean

R10-5-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi1() เป็น void ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ voidclassi1

R10-6) จากข้อที่ R10-5 ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Opi1() ในแอ็บสแตร็คแมชชีนบี Classi ดังนี้

R10-6-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi1() เป็น STRING หรือ void ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Opi1() ในอนุประโยค THEN ของ

แอ็บสแตร็คแมชชีนบี Classi คือ classi1 := null หรือ voidclassi1 := null ตามลำดับ

R10-6-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi1() เป็น BOOLEAN ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Opi1() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ boolean := FALSE

R10-6-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi1() เป็น NAT ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Opi1() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ classi1 := 0

R10-7) กำหนดอนุประโยค END เพื่อจบการทำงานของอนุประโยค OPERATIONS

Opi2() เป็นโอเปอเรชันนอนเบซิคที่เรียกโอเปอเรชันเบซิค Opj() ภายใน Classi

R10-8) กำหนดอนุประโยค MACHINE

R10-9) กำหนดแอ็บสแตร็คแมชชีนบี IntermediateClassi ในอนุประโยค MACHINE

R10-10) กำหนดอนุประโยค USES

R10-11) ระบุชื่อของแอ็บสแตร็คแมชชีนบี คือ BasicClassi ในอนุประโยค USES

R10-12) กำหนดอนุประโยค OPERATIONS

R10-13) ระบุโอเปอเรชัน Intermediate_Opi2() ในอนุประโยค OPERATIONS โดยพิจารณาข้อที่ R10-14 และ R10-15

R10-14) พิจารณาโอเปอเรชัน Intermediate_Opi2() มีพารามิเตอร์ของโอเปอเรชัน Intermediate_Opi2() หรือไม่ ดังนี้

R10-14-1) ถ้าโอเปอเรชัน Intermediate_Opi2() มีพารามิเตอร์ ให้กำหนดอนุประโยค PRE โดยทำการระบุเงื่อนไขของความสัมพันธ์ระหว่างพารามิเตอร์และชนิดของพารามิเตอร์ นั่นคือ Parameteri2 ∈ TypeParami2 แล้วกำหนดอนุประโยค THEN

R10-14-2) ถ้าโอเปอเรชัน Intermediate_Opi2() ไม่มีพารามิเตอร์ ให้กำหนดอนุประโยค BEGIN

R10-15) พิจารณาชนิดค่าที่ส่งคืนของโอเปอเรชัน Intermediate_Opi2() ดังนี้

R10-15-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Intermediate_Opi2() เป็น STRING หรือ NAT ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี IntermediateClassi คือ intermediateclassi1

R10-15-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Intermediate_Opi2() เป็น BOOLEAN ทำ

การระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี IntermediateClassi คือ boolean

R10-15-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Intermediate_Opi2() เป็น void ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี IntermediateClassi คือ voidintermediateclassi1

R10-16) จากข้อที่ R10-15 ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Intermediate_Opi2() ในแอ็บสแตร็คแมชชีนบี IntermediateClassi ดังนี้

R10-16-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Intermediate_Opi2() เป็น STRING หรือ void ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Intermediate_Opi2() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี IntermediateClassi คือ intermediateclassi1 := null หรือ voidintermediateclassi1 := null ตามลำดับ

R10-16-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Intermediate_Opi2() เป็น BOOLEAN ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Intermediate_Opi2() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี IntermediateClassi คือ boolean := FALSE

R10-16-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Intermediate_Opi2() เป็น NAT ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Intermediate_Opi2() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี IntermediateClassi คือ intermediateclassi1 := 0

R10-17) กำหนดอนุประโยค END เพื่อจบการทำงานของอนุประโยค OPERATIONS

R10-18) กำหนดอนุประโยค END เพื่อจบการทำงานของแอ็บสแตร็คแมชชีนบี IntermediateClassi

Opj() เป็นโอเปอเรชันเบซิค

R10-19) พิจารณาแอ็บสแตร็คแมชชีนบี BasicClassj

R10-20) กำหนดอนุประโยค OPERATIONS

R10-21) ระบุโอเปอเรชัน Basic_Opj() ในอนุประโยค OPERATIONS โดยพิจารณาข้อที่ R10-22 และ R10-23

R10-22) พิจารณาโอเปอเรชัน Basic_Opj() มีพารามิเตอร์ของโอเปอเรชัน Basic_Opi() หรือไม่ ดังนี้

R10-22-1) ถ้าโอเปอเรชัน Basic_Opj() มีพารามิเตอร์ ให้กำหนดอนุประโยค PRE โดยทำการระบุเงื่อนไขของความสัมพันธ์ระหว่างพารามิเตอร์และชนิดของพารามิเตอร์

นั่นคือ Parameterj ∈ TypeParamj แล้วกำหนดอนุประโยค THEN

R10-22-2) ถ้าโอเปอเรชัน Basic_Opj() ไม่มีพารามิเตอร์ ให้กำหนดอนุประโยค BEGIN

R10-23) พิจารณาชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() ดังนี้

R10-23-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Opi() เป็น STRING หรือ NAT ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicClassj คือ basicclassj1

R10-23-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() เป็น BOOLEAN ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicClassj คือ boolean

R10-23-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() เป็น void ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicClassj คือ voidbasicclassj1

R10-24) จากข้อที่ R10-23 ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Basic_Opj() ในแอ็บสแตร็คแมชชีนบี BasicClassj ดังนี้

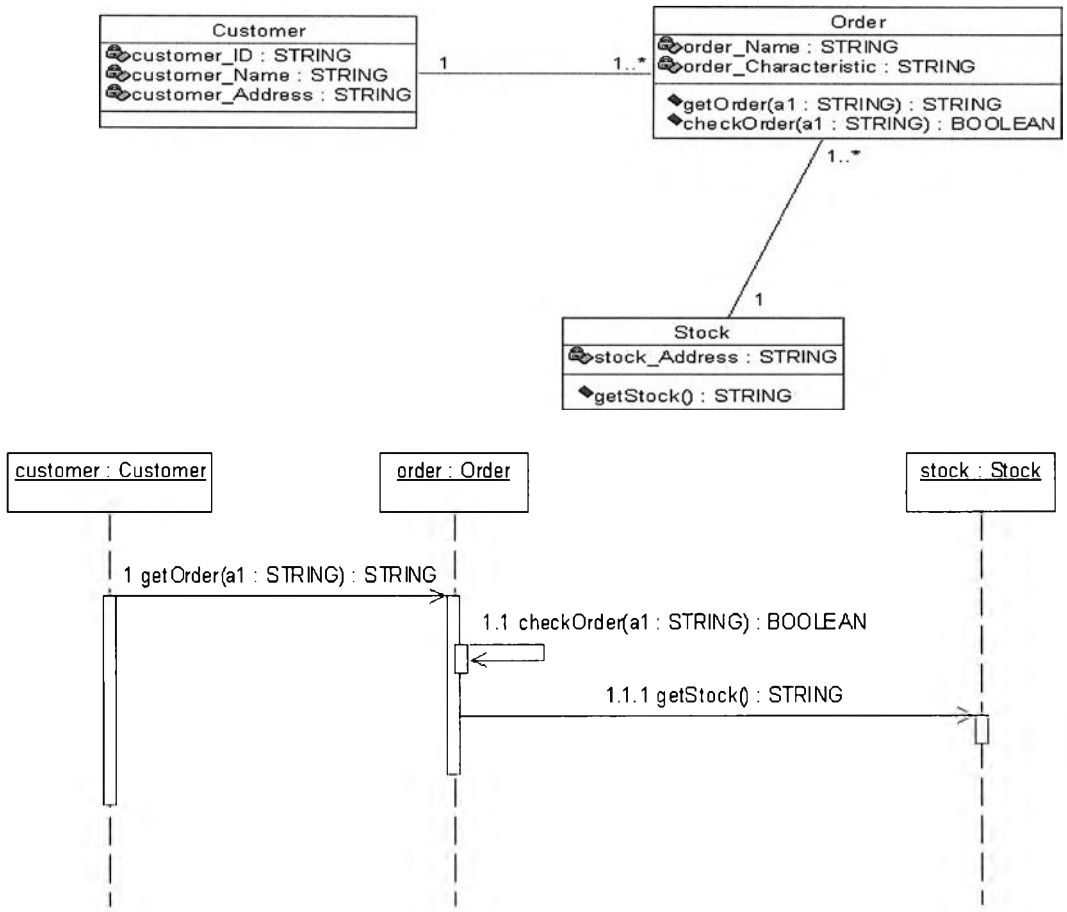
R10-24-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() เป็น STRING หรือ void ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Basic_Opj() ในอนุประโยค THENของแอ็บสแตร็คแมชชีนบี BasicClassj คือ basicclassj1 := null หรือ voidbasicclassj1 := null ตามลำดับ

R10-24-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() เป็น BOOLEAN ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Basic_Opj() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicClassj คือ boolean := FALSE

R10-24-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_Opj() เป็น NAT ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Basic_Opj() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicClassj คือ basicclassj1 := 0

R10-25) กำหนดอนุประโยค END เพื่อจบการทำงานของอนุประโยค OPERATIONS

จากตัวอย่างของแผนภาพคลาสที่แสดงส่วนหนึ่งของระบบการสั่งซื้อสินค้า ซึ่งประกอบไปด้วยคลาส Customer คลาส Order และคลาส Stock รวมทั้งแผนภาพซีเควนซ์ที่แสดงเหตุการณ์ของการสั่งซื้อสินค้าจากลูกค้า แสดงได้ดังรูปที่ 3.85



รูปที่ 3.85 ตัวอย่างแผนภาพคลาสและซีเควนซ์ของเหตุการณ์การสั่งซื้อสินค้าที่อธิบายในกฎข้อที่ 10

จากรูปที่ 3.85 จะพบว่า คลาส Order จะประกอบไปด้วยโอเปอเรชัน getOrder(a1 : STRING) : STRING และโอเปอเรชัน checkOrder(a1 : STRING) : BOOLEAN ส่วนคลาส Stock มีโอเปอเรชัน getStock( ) : STRING และจากแผนภาพซีเควนซ์ที่แสดงเหตุการณ์การสั่งซื้อสินค้าจากลูกค้า จะพบว่ามีการรับส่งข้อความระหว่างออบเจ็ค customer ของคลาส Customer กับออบเจ็ค order ของคลาส Order และมีการรับส่งข้อความระหว่างออบเจ็ค order ของคลาส Order กับออบเจ็ค stock ของคลาส Stock เมื่อพิจารณาโอเปอเรชันของคลาส Stock จะพบว่าโอเปอเรชัน getStock( ) เป็นโอเปอเรชันเบซิค สามารถทำการระบุโอเปอเรชัน Basic_getStock ( ) ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี BasicStock แสดงได้ดังรูปที่ 3.86

ส่วนโอเปอเรชันของคลาส Order จะพบว่าโอเปอเรชัน getOrder( ) และโอเปอเรชัน checkOrder( ) เป็นโอเปอเรชันนอนเบซิค สามารถทำการระบุโอเปอเรชัน getOrder( ) ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี Order ส่วนโอเปอเรชันนอนเบซิค checkOrder( ) จะเป็นโอเปอเรชันที่ถูกเรียกจากโอเปอเรชัน getOrder( ) ภายในคลาสเดิมทำการสร้างแอ็บสแตร็คแมชชีนบี IntermediateOrder และระบุโอเปอเรชัน Intermediate_checkOrder( ) ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี IntermediateOrder แสดงได้ดังรูปที่ 3.87 และรูปที่ 3.88

```
MACHINE
BasicStock
...
OPERATIONS
basicstock1 ← Basic_getStock =
BEGIN
...
END

END
```

รูปที่ 3.86 ระบุโอเปอเรชัน Basic_getStock() ในแอ็บสแตร็คแมชชีนบี BasicStock

```
MACHINE
Order
...
OPERATIONS
order1 ← getOrder(a1) =
PRE
a1 ∈ STRING
THEN
...
END

END
```

รูปที่ 3.87 ระบุโอเปอเรชัน getOrder() ในแอ็บสแตร็คแมชชีนบี Order

```
MACHINE
IntermediateOrder
...
OPERATIONS
boolean ← Intermedaite_checkOrder(a1) =
PRE
a1 ∈ STRING
THEN
...
END

END
```

รูปที่ 3.88 สร้างแอ็บสแตร็คแมชชีนบี IntermediateOrder ทำการระบุโอเปอเรชัน Intermediate_checkOrder( )

3.3.2 การถ่ายทอดโอเปอเรชันจากซูเปอร์คลาส

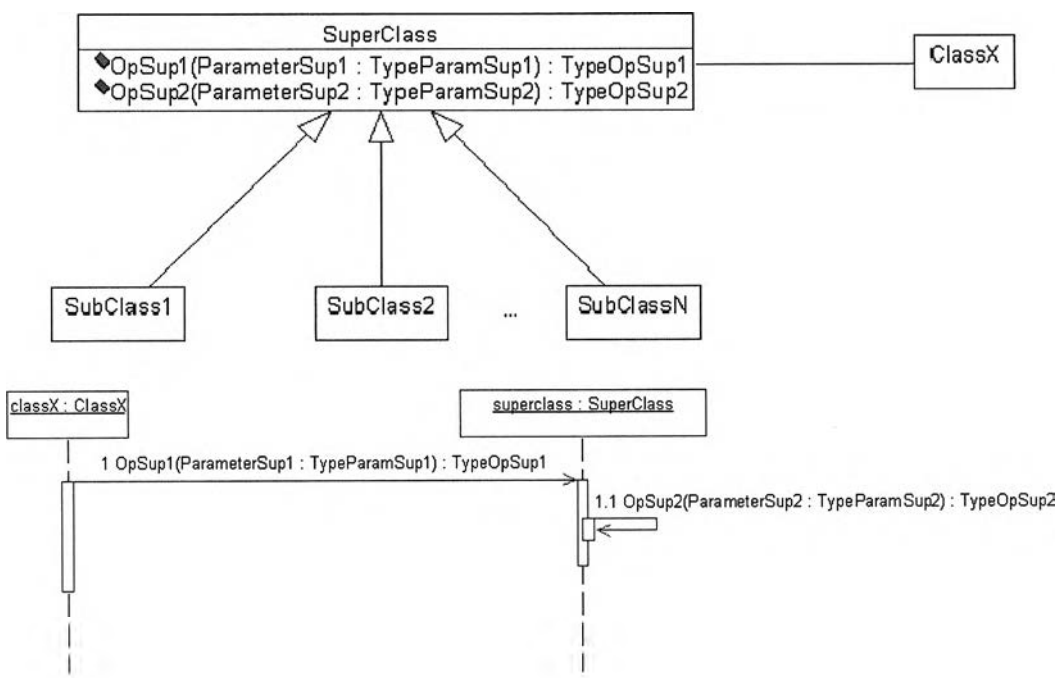
จากคุณสมบัติของการถ่ายทอดโอเปอเรชันจากซูเปอร์คลาส เป็นการที่ซับคลาสมีโอเปอเรชันทั้งหมดจากซูเปอร์คลาส และในขณะเดียวกันก็สามารถมีโอเปอเรชันของซับคลาสเองได้ สามารถทำให้ซับคลาสที่ได้รับการถ่ายทอดโอเปอเรชันจากซูเปอร์คลาสนั้น มีความสามารถเฉพาะที่ซูเปอร์คลาสไม่สามารถทำได้ ทำการระบุโอเปอเรชันที่ทำการถ่ายทอดจากซูเปอร์คลาสทั้งหมดในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบีของซับคลาส โดยการนำเสนอในกฎข้อที่ 11 นั่นคือ

กฎข้อที่ 11

การระบุโอเปอเรชันในแอ็บสแตร็คแมซชีนบีซับคลาส แอ็บสแตร็คแมซชีนบีอินเทอร์มิเดียทซับคลาส และเบซิคซับคลาสโดยอาศัยคุณสมบัติการถ่ายทอดโอเปอเรชันจากซูเปอร์คลาส

จากความสัมพันธ์เจนเนอรัลไลเซชันในเชิงของการถ่ายทอดโอเปอเรชันทั้งหมดมาจากซูเปอร์คลาสให้ซับคลาส สามารถระบุโอเปอเรชันเหล่านี้ในแอ็บสแตร็คแมซชีนบีซับคลาส แอ็บสแตร็คแมซชีนบีอินเทอร์มิเดียทซับคลาส และแอ็บสแตร็คแมซชีนบีเบซิคซับคลาส โดยอาศัยแนวคิดจากการสร้างแอ็บสแตร็คแมซชีนบีซับคลาสที่รับถ่ายทอดคุณลักษณะมาจากซูเปอร์คลาสดังที่ได้กล่าวไว้โดยการระบุคุณลักษณะทั้งหมดมาจากซูเปอร์คลาสในอนุประโยค VARIABLES ที่ได้กล่าวไว้ในกฎข้อที่ 7

ในกฎข้อนี้จะทำการระบุโอเปอเรชันในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมซชีนบีซับคลาส แอ็บสแตร็คแมซชีนบีอินเทอร์มิเดียทซับคลาส และแอ็บสแตร็คแมซชีนบีเบซิคซับคลาส โดยอาศัยคุณสมบัติการถ่ายทอดโอเปอเรชันทั้งหมดจากซูเปอร์คลาส พิจารณาจากแผนภาพคลาสและแผนภาพซีเควนซ์ แสดงได้ดังรูปที่ 3.89



รูปที่ 3.89 รูปแบบแผนภาพคลาสและแผนภาพซีเควนซ์ที่อธิบายในกฎข้อที่ 11

จากรูปที่ 3.89 ซูเปอร์คลาส SuperClass จะมีความสัมพันธ์กับ ClassX ซึ่งจะพบในแผนภาพคลาส และมีการส่งข้อความระหว่างออบเจ็ค classX ของคลาส ClassX กับออบเจ็ค superclass ของคลาส SuperClass ในแผนภาพซีเควนซ์ โดย SuperClass จะมีโอเปอเรชัน คือ OpSup1(ParameterSup1 : TypeParamSup1) : TypeOpSup1 และโอเปอเรชัน

OpSup2(ParameterSup2 : TypeParamSup2) : TypeOpSup2 ส่วนซับคลาส ได้แก่ SubClass1 ถึง SubClassN จะไม่มีโอเปอเรชัน

ทำการระบุโอเปอเรชันของ SuperClass ในแอ็บสแตร็คแมชชีนบีของซับคลาสทั้งหมดโดยอาศัยคุณสมบัติการถ่ายทอดโอเปอเรชันจากซูเปอร์คลาส นั่นคือแอ็บสแตร็คแมชชีนบีของซับคลาส ได้แก่ SubClass1 ถึง SubClassN จะมีโอเปอเรชัน OpSup1( ) และโอเปอเรชัน OpSup2( ) ที่ถ่ายทอดจาก SuperClass โดยโอเปอเรชัน OpSup1( ) เป็นโอเปอเรชันนอนเบซิค และโอเปอเรชัน OpSup2( )เป็นโอเปอเรชันเบซิค ดังนั้นทำการระบุโอเปอเรชัน OpSup1 ( ) ในแอ็บสแตร็คแมชชีนบีซูเปอร์คลาสรวมทั้งในแอ็บสแตร็คแมชชีนบีซับคลาสทั้งหมด และทำการระบุโอเปอเรชัน OpSup2 ( ) ในแอ็บสแตร็คแมชชีนบีเบซิคซูเปอร์คลาส รวมทั้งในแอ็บสแตร็คแมชชีนบีเบซิคซับคลาสทั้งหมด แสดงได้ดังรูปที่ 3.90 ถึงรูปที่ 3.93

```
MACHINE
SuperClass
...
USES
BasicSuperClass
OPERATIONS
superclass1 ← OpSup1(ParameterSup1) =
PRE
ParameterSup1 ∈ TypeParamSup1
THEN
...
END

END
```

รูปที่ 3.90 ระบุโอเปอเรชัน OpSup1() ในแอ็บสแตร็คแมชชีนบี SuperClass

```
MACHINE
BasicSuperClass
...
OPERATIONS
basicsuperclass1 ← Basic_OpSup2(ParameterSup2) =
PRE
ParameterSup2 ∈ TypeParamSup2
THEN
...
END

END
```

รูปที่ 3.91 ระบุโอเปอเรชัน Basic_OpSup2() ในแอ็บสแตร็คแมชชีนบี BasicSuperClass

```
MACHINE
SubClass1
...
USES
BasicSubClass1
OPERATIONS
subclass11 ← OpSup1(ParameterSup1) =
PRE
ParameterSup1 ∈ TypeParamSup1
THEN
...
END

END
```

```
MACHINE
SubClass2
...
USES
BasicSubClass2
OPERATIONS
subclass21 ← OpSup1(ParameterSup1) =
PRE
ParameterSup1 ∈ TypeParamSup1
THEN
...
END

END
```

...

```
MACHINE
SubClassN
...
USES
BasicSubClassN
OPERATIONS
subclassN1 ← OpSup1(ParameterSup1) =
PRE
ParameterSup1 ∈ TypeParamSup1
THEN
...
END

END
```

รูปที่ 3.92 ระบุโอเปอเรชัน OpSup1() ในแอ็บสแตร็คแมชชีนบี SubClass1 ถึง SubClassN

```
MACHINE
BasicSubClass1
...
OPERATIONS
basicsubclass11 ← Basic_OpSup2(ParameterSup2) =
PRE
ParameterSup2 ∈ TypeParamSup2
THEN
...
END

END
```

```
MACHINE
BasicSubClass2
...
OPERATIONS
basicsubclass21 ← Basic_OpSup2(ParameterSup2) =
PRE
ParameterSup2 ∈ TypeParamSup2
THEN
...
END

END
```

...

```
MACHINE
BasicSubClassN
...
OPERATIONS
basicsubclassN1 ← Basic_OpSup2(ParameterSup2) =
PRE
ParameterSup2 ∈ TypeParamSup2
THEN
...
END

END
```

รูปที่ 3.93 ระบุโอเปอเรชัน Basic_OpSup2() ในแอ็บสแตร็คแมชชีนบี BasicSubClass1 ถึง BasicSubClassN

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 11 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 11

สำหรับ SuperClassi ใด ๆ ในแผนภาพคลาส โดยที่ SuperClassi ใด ๆ จะถ่ายทอดโอเปอเรชันทั้งหมดนี้ให้กับ SubClassi ซึ่ง SubClassi จะมีโอเปอเรชันทั้งหมดที่มาจาก SuperClassi คือ OpSupi1() : TypeOpSupi1และ OpSupi2() : TypeOpSupi2 พิจารณาในแผนภาพคลาสจะพบว่า ClassX มีความสัมพันธ์กับ SuperClass เมื่อพิจารณาเหตุการณ์ที่ k ในแผนภาพซีเควนซ์

ออบเจ็ค classX ของคลาส ClassX จะมีการร้องขอให้ออบเจ็ค superclass ของคลาส SuperClass ทำโอเปอเรชัน OpSupi1() : TypeOpSupi1 ออบเจ็ค superclass ของคลาส SuperClass จะทำโอเปอเรชัน OpSupi2() : TypeOpSupi2 โดยที่โอเปอเรชัน OpSupi1() จะเป็นโอเปอเรชันที่เรียกโอเปอเรชัน OpSupi2()

จากข้อความข้างต้นจะสามารถทำการระบุโอเปอเรชัน OpSupi1() และโอเปอเรชัน OpSupi2() จากแผนภาพคลาสที่แสดงในเหตุการณ์ที่ k ของแผนภาพซีเควนซ์ในแอ็บสแตร็คแมชชีนบี ดังนี้

OpSupi1() เป็นโอเปอเรชันนอนเบซิค

R11-1) พิจารณาแอ็บสแตร็คแมชชีนบี SubClassi

R11-2) กำหนดอนุประโยค OPERATIONS

R11-3) ระบุโอเปอเรชัน OpSupi1() ในอนุประโยค OPERATIONS โดยพิจารณาข้อที่ R11-4 และ R11-5

R11-4) พิจารณาโอเปอเรชัน OpSupi1() มีพารามิเตอร์ของโอเปอเรชัน OpSupi1() หรือไม่ ดังนี้

R11-4-1) ถ้าโอเปอเรชัน OpSupi1() มีพารามิเตอร์ ให้กำหนดอนุประโยค PRE โดยทำการระบุเงื่อนไขของความสัมพันธ์ระหว่างพารามิเตอร์และชนิดของพารามิเตอร์ นั่นคือ ParameterSupi1 ∈ TypeParamSupi1 แล้วกำหนดอนุประโยค THEN

R11-4-2) ถ้าโอเปอเรชัน OpSupi1() ไม่มีพารามิเตอร์ ให้กำหนดอนุประโยค BEGIN

R11-5) พิจารณาชนิดค่าที่ส่งคืนของโอเปอเรชัน OpSupi1() ดังนี้

R11-5-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน OpSupi1() เป็น STRING หรือ NAT ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี Classi คือ classi1

R11-5-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน OpSupi1() เป็น BOOLEAN ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี SubClassi คือ boolean

R11-5-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน OpSupi1() เป็น void ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี SubClassi คือ voidsubclassi1

R11-6) จากข้อที่ R11-5 ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน OpSupi1() ในแอ็บสแตร็คแมชชีนบี SubClassi ดังนี้

R11-6-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน OpSupi1() เป็น STRING หรือ void ทำการ

ระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน OpSupi1() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี SubClassi คือ subclassi1 := null หรือ voidsubclassi1 := null ตามลำดับ

R11-6-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน OpSupi1() เป็น BOOLEAN ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน OpSupi1() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี SubClassi คือ boolean := FALSE

R11-6-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน OpSupi1() เป็น NAT ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน OpSupi1() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี SubClassi คือ subclassi1 := 0

R11-7) กำหนดอนุประโยค END เพื่อจบการทำงานของอนุประโยค OPERATIONS

OpSupi2() เป็นโอเปอเรชันเบซิค

R11-8) พิจารณาแอ็บสแตร็คแมชชีนบี BasicSubClassi

R11-9) กำหนดอนุประโยค OPERATIONS

R11-10) ระบุโอเปอเรชัน Basic_OpSupi2() ในอนุประโยค OPERATIONS โดยพิจารณาข้อที่ R11-11 และ R11-12

R11-11) พิจารณาโอเปอเรชัน Basic_OpSupi2() มีพารามิเตอร์ของโอเปอเรชัน Basic_OpSupi2() หรือไม่ ดังนี้

R11-11-1) ถ้าโอเปอเรชัน Basic_OpSupi2() มีพารามิเตอร์ ให้กำหนดอนุประโยค PRE โดยทำการระบุเงื่อนไขของความสัมพันธ์ระหว่างพารามิเตอร์และชนิดของพารามิเตอร์ นั่นคือ ParameterSupi2 ∈ TypeParamSupi2 แล้วกำหนดอนุประโยค THEN

R11-11-2) ถ้าโอเปอเรชัน Basic_OpSupi2() ไม่มีพารามิเตอร์ ให้กำหนดอนุประโยค BEGIN

R11-12) พิจารณาชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_OpSupi2() ดังนี้

R11-12-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_OpSupi2() เป็น STRING หรือ NAT ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicSubClassi คือ basicsubclassi1

R11-12-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_OpSupi2() เป็น BOOLEAN ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicSubClassi คือ boolean

R11-12-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_OpSupi2() เป็น void ทำการระบุตัวแปรผลลัพธ์ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicSubClassi คือ voidbasicsubclassi1

R11-13) จากข้อที่ R11-12 ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Basic_OpSupi2() ในแอ็บสแตร็คแมชชีนบี BasicSubClassi ดังนี้

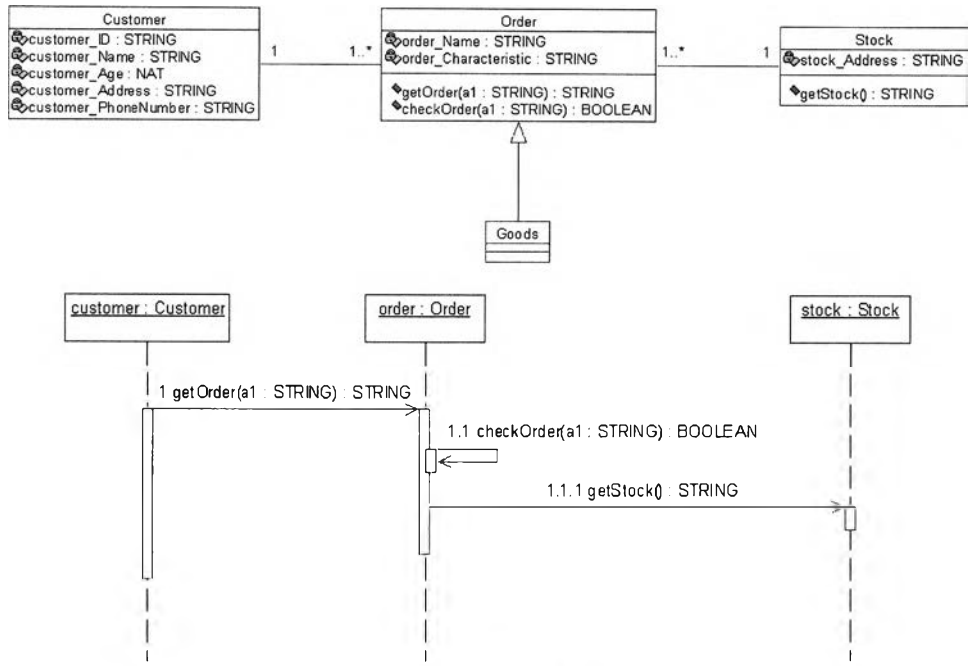
R11-13-1) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_OpSupi2() เป็น STRING หรือ void ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Basic_OpSupi2() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicSubClassi คือ basicsubclassi1 := null หรือ voidbasicsubclassi21 := null ตามลำดับ

R11-13-2) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_OpSupi2() เป็น BOOLEAN ทำการระบุค่าเริ่มต้นของตัวแปรผลสัพธ์ของโอเปอเรชัน Basic_OpSupi2() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicSubClassi คือ boolean := FALSE

R11-13-3) ถ้าชนิดค่าที่ส่งคืนของโอเปอเรชัน Basic_OpSupi2() เป็น NAT ทำการระบุค่าเริ่มต้นของตัวแปรผลลัพธ์ของโอเปอเรชัน Basic_OpSupi2() ในอนุประโยค THEN ของแอ็บสแตร็คแมชชีนบี BasicSubClassi คือ basicsubclassi1 := 0

R11-14) กำหนดอนุประโยค END เพื่อจบการทำงานของอนุประโยค OPERATIONS

จากตัวอย่างของแผนภาพคลาสที่แสดงส่วนหนึ่งของระบบการสั่งซื้อสินค้า ซึ่งประกอบไปด้วย คลาส Customer คลาส Order คลาส Goods และคลาส Stock รวมทั้งแผนภาพซีเควนซ์ที่แสดงเหตุการณ์การสั่งซื้อสินค้า พิจารณาแผนภาพคลาสจะพบว่าคลาส Order ซึ่งเป็นซูเปอร์คลาสถ่ายทอดโอเปอเรชันทั้งหมดให้กับคลาส Goods ซึ่งเป็นซับคลาส คลาส Order มีความสัมพันธ์แอสโซซิเอชันกับคลาส Customer และคลาส Stock ในตัวอย่างนี้จะพบว่าคลาส Order จะมีโอเปอเรชัน getOrder(a1 : STRING) : STRING และโอเปอเรชัน checkOrder(a1 : STRING) : BOOLEAN นั่นคือคลาส Goods ซึ่งเป็นซับคลาส จะได้รับการถ่ายทอดโอเปอเรชันทั้งหมดจากซูเปอร์คลาส Order โดยคลาส Goods จะมีโอเปอเรชัน getOrder( ) และโอเปอเรชัน checkOrder( ) จากซูเปอร์คลาส Order และพิจารณาแผนภาพซีเควนซ์ของเหตุการณ์การสั่งซื้อสินค้าจากลูกค้า แสดงได้ดังรูปที่ 3.94



รูปที่ 3.94 ตัวอย่างแผนภาพคลาสและซีเควนซ์ระบบการสั่งซื้อสินค้า โดยคลาส Goods เป็นซับคลาส

จากรูปที่ 3.94 แอ็บสแตร็คแมชชีนบี Goods จะอาศัยคุณสมบัติการถ่ายทอดโอเปอเรชันจากซูเปอร์คลาส นั่นคือ ทำการระบุโอเปอเรชัน getOrder( ) ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี Goods และ ทำการสร้างแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาส คือ IntermediateGoods พร้อมทั้งระบุโอเปอเรชัน Intermediate_checkOrder( ) ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี IntermediateGoods แสดงได้ดังรูปที่ 3.95 และรูปที่ 3.96

```
MACHINE
Goods
...
OPERATIONS
goods1 ← getOrder(a1) =
PRE
a1 ∈ STRING
THEN
...
END

END
```

รูปที่ 3.95 แอ็บสแตร็คแมชชีนบี Goods มีโอเปอเรชัน getOrder() ถ่ายทอดจากซูเปอร์คลาส

```
MACHINE
IntermediateGoods
...
OPERATIONS
intermediategoods1 ← Intermediate_checkOrder(a1) =
PRE
a1 ∈ STRING
THEN
...
END

END
```

รูปที่ 3.96 สร้างแอ็บสแตร็คแมชชีนบี IntermediateGoods มีโอเปอเรชัน Intermediate_checkOrder() ที่ถ่ายทอดจากซูเปอร์คลาส

3.3.3 การสร้างอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี และการสร้างอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีซับคลาส เพื่อแสดงการเรียกโอเปอเรชันของคลาสที่แสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คจากเหตุการณ์ต่าง ๆ ของแผนภาพซีเควนซ์ รวมทั้งการระบุเงื่อนไขที่เป็นจริงของโอเปอเรชันในเหตุการณ์จากแผนภาพซีเควนซ์

การเรียกโอเปอเรชันของคลาสที่แสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คจากเหตุการณ์ของแผนภาพซีเควนซ์นั้น จะเป็นการที่โอเปอเรชันที่เรียกจะเรียกโอเปอเรชันที่ถูกเรียกที่อยู่ในคลาสเดียวกันหรือคลาสอื่นๆ ให้สามารถกระทำโอเปอเรชันที่ถูกเรียกตามที่ต้องการได้ โดยแสดงในรูปแบบของการอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาสที่มีโอเปอเรชันที่เรียก เพื่อแสดงให้เห็นถึงการเรียกโอเปอเรชัน นำเสนอในกฎข้อที่ 12 กฎข้อที่ 13 และกฎข้อที่ 14 นั่นคือ

กฎข้อที่ 12

การสร้างอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีเพื่อแสดงการเรียกโอเปอเรชันจากเหตุการณ์ของแผนภาพซีเควนซ์

แอ็บสแตร็คแมชชีนบีคลาส และแอ็บสแตร็คแมชชีนบีอิมเทอร์มิเดียทคลาสสามารถนำมาสร้างเป็นอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาส และสร้างอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสที่มีโอเปอเรชันที่เรียก เพื่อแสดงการเรียกโอเปอเรชันของคลาสที่แสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คภายในเหตุการณ์ต่าง ๆ ของแผนภาพซีเควนซ์ โดยที่รูปแบบของอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาส คือ Class_imp และอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีอิมเทอร์มิเดียทคลาส คือ IntermediateClass_ imp ที่มีโอเปอเรชันที่เรียก แสดงได้ดังรูปที่ 3.97 และรูปที่ 3.98

IMPLEMENTATION
Class_imp
REFINES
BAM of Class which has Calling Operations
SEES
Library BAM
BAM of Class or IntermediateClass or BasicClass which has Called Operations
OPERATIONS
Output Variable of Calling Operations from BAM of Class which has Calling Operations
← Calling Operations (Parameter) =
VAR
Output Variable of Called Operations from BAM of Class or Intermediate Class or Basic Class which has Called Operations
IN
Output Variable of Called Operations from BAM of Class or Intermediate Class or Basic Class which has Called Operations
← Called Operations (Parameter)
...
END

END

รูปที่ 3.97 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี Class_imp

IMPLEMENTATION
IntermediateClass_imp
REFINES
BAM of IntermediateClass which has Calling Operations
SEES
Library BAM
BAM of Class or IntermediateClass or BasicClass which has Called Operations
OPERATIONS
Output Variable of Calling Operations from BAM of IntermediateClass which has Calling Operations
←Calling Operations (Parameter) =
VAR
Output Variable of Called Operations from BAM of Class or Intermediate Class or Basic Class which has Called Operations
IN
Output Variable of Called Operations from BAM of Class or Intermediate Class or Basic Class which has Called Operations
← Called Operations (Parameter)
...
END

END

รูปที่ 3.98 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี IntermediateClass_imp

โดยที่ Class_imp

คือ ชื่ออิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาสที่มีโอเปอเรชันที่เรียก

IntermediateClass_imp

คือ ชื่อของอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสที่มีโอเปอเรชันที่เรียก

Library BAM

คือ ไลบรารีแอ็บสแตร็คแมชชีนบี StringType และ BooleanType

BAM of Class which has Calling Operations

คือ แอ็บสแตร็คแมชชีนบีคลาสที่มีโอเปอเรชันที่เรียก

BAM of IntermediateClass which has Calling Operations

คือ แอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสที่มีโอเปอเรชันที่เรียก

BAM of Class or IntermediateClass or BasicClass which has Called Operations

คือ แอ็บสแตร็คแมชชีนบีคลาสหรือแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาส หรือแอ็บสแตร็คแมชชีนบีเบซิคคลาสที่มีโอเปอเรชันที่ถูกเรียก

Calling Operations (Parameter)

คือ โอเปอเรชันที่เรียก โดยมีการระบุตัวแปรเข้า (Input Variables) หรือพารามิเตอร์

Output Variable of Calling Operations from BAM of Class which has Calling Operations

คือ ค่าของตัวแปรผลลัพธ์ (Output Variables) หรือชนิดค่าที่ส่งคืนของโอเปอเรชันจากแอ็บสแตร็คแมชชีนบีคลาสที่มีโอเปอเรชันที่เรียก

Called Operations (Parameter)

คือ โอเปอเรชันที่ถูกเรียก โดยมีการระบุตัวแปรเข้า หรือพารามิเตอร์

Output Variable of Calling Operations from BAM of IntermediateClass which has Calling Operations

คือ ค่าของตัวแปรผลลัพธ์ หรือชนิดค่าที่ส่งคืนของโอเปอเรชันจากแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสที่มีโอเปอเรชันที่เรียก

Output Variable of Called Operations from BAM of Class or Intermediate Class or Basic Class which has Called Operations

คือ ค่าของตัวแปรผลลัพธ์ หรือชนิดค่าที่ส่งคืนของโอเปอเรชันจากแอ็บสแตร็คแมชชีนบีคลาส หรือจากแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาส หรือแอ็บสแตร็คแมชชีนบีเบซิคคลาสที่มีโอเปอเรชันที่ถูกเรียก

จากรูปที่ 3.81 จะพบว่าในแผนภาพคลาสจะประกอบไปด้วย ClassA ClassB และ ClassC และจากแผนภาพซีเควนซ์ที่แสดงเหตุการณ์ของการรับส่งข้อความระหว่างออบเจ็ค พบว่าโอเปอเรชัน OpB1(ParameterB1 : TypeParamB1) : TypeOpB1 เรียกโอเปอเรชัน

OpB2(ParameterB2 : TypeParamB2) : TypeOpB2 และโอเปอเรชัน OpB2(ParameterB2 : TypeParamB2) : TypeOpB2 เรียกโอเปอเรชัน OpC1(ParameterC1 : TypeParamC1) : TypeOpC1 ทำให้ได้ข้อสรุปเพื่อทำการสร้างอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี ดังต่อไปนี้

1. โอเปอเรชัน OpB1( ) เป็นโอเปอเรชันนอนเบซิค ทำการระบุโอเปอเรชันนี้ในแอ็บสแตร็คแมชชีนบี ClassB โดยที่โอเปอเรชันOpB1( ) เป็นโอเปอเรชันที่เรียกโอเปอเรชัน OpB2( )

2. โอเปอเรชัน OpB2( ) เป็นโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชัน OpB1( ) ภายในคลาสเดิม ทำการสร้างแอ็บสแตร็คแมชชีนบี IntermediateClassB และระบุโอเปอเรชันนี้ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี IntermediateClassB ใช้คำนำหน้า Intermediate แล้วตามด้วยโอเปอเรชัน OpB2( ) นั่นคือ Intermediate_OpB2()

3. โอเปอเรชัน OpC1( ) เป็นโอเปอเรชันเบซิคที่ถูกเรียกจากโอเปอเรชันนอนเบซิค OpB2( ) ทำการระบุโอเปอเรชันนี้ในแอ็บสแตร็คแมชชีนบี BasicClassC ใช้คำนำหน้า Basic แล้วตามด้วยโอเปอเรชัน OpC1( ) นั่นคือ Basic_OpC1()

แสดงอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบี ClassB_imp และอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบี IntermediateClassB_imp ได้ดังรูปที่ 3.99 และรูปที่ 3.100

```
IMPLEMENTATION
ClassB_imp
REFINES
ClassB
SEES
...
IntermediateClassB
OPERATIONS
classb1 ← OpB1(ParameterB1) =
VAR
intermediateclassb1
IN
intermediateclassb1 ← Intermediate_OpB2(ParameterB2)
END

END
```

รูปที่ 3.99 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี ClassB_imp

```
IMPLEMENTATION
IntermediateClassB_imp
REFINES
IntermediateClassB
SEES
...
BasicClassC
OPERATIONS
intermediateclassb1 ← Intermediate_OpB2(ParameterB2) =
VAR
basicclassc1
IN
basicclassc1 ← Basic_OpC1(ParameterC1)
END

END
```

รูปที่ 3.100 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี IntermediateClassB_imp

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 12 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 12

สำหรับ Classi Classj และ ClassX ในแผนภาพคลาส โดยที่ Classi จะมีโอเปอเรชัน Opi1() : TypeOpi1 และ Opi2() : TypeOpi2 ส่วน Classj จะมีโอเปอเรชัน Opj() : TypeOpj พิจารณาในแผนภาพคลาสจะพบว่า ClassX จะมีความสัมพันธ์กับ Classi และ Classi จะมีความสัมพันธ์กับ Classj จะสามารถแสดงได้ในเหตุการณ์ที่ k ของแผนภาพซีเควนซ์ นั่นคือ ออบเจ็ค classX ของคลาส ClassX จะมีการร้องขอให้ออบเจ็ค classi ของคลาส Classi ทำโอเปอเรชัน Opi1() ซึ่งออบเจ็ค classi ของคลาส Classi จะทำโอเปอเรชัน Opi2() และออบเจ็ค classi ของคลาส Classi จะมีการร้องขอให้ออบเจ็ค classj ของคลาส Classj ทำโอเปอเรชัน Opj() โดยที่ โอเปอเรชัน Opi1() จะเป็นโอเปอเรชันที่เรียกโอเปอเรชัน Opi2() และโอเปอเรชัน Opi2() จะเป็นโอเปอเรชันที่เรียกโอเปอเรชัน Opj()

จากข้อความข้างต้นจะสามารถสรุปได้ว่า

1. โอเปอเรชัน Opi1() เป็นโอเปอเรชันนอนเบซิค
2. โอเปอเรชัน Opi2() เป็นโอเปอเรชันนอนเบซิคที่เรียกภายใน Classi และ
3. โอเปอเรชัน Opj() เป็นโอเปอเรชันเบซิค

จากข้อสรุปทั้ง 3 สามารถนำมาสร้างอิมพลีเทชันแอ็บสแตร็คแมชชีนบี Classi นั่นคือ Classi_imp และอิมพลีเทชันแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียท Classi นั่นคือ IntermediateClassi_imp ดังนี้

Opi1() เป็นโอเปอเรชันนอนเบซิคที่เรียกโอเปอเรชัน Opi2()

R12-1) กำหนดอนุประโยค IMPLEMENTATION

R12-2) กำหนด Classi_imp ในอนุประโยค IMPLEMENTATION

R12-3) กำหนดอนุประโยค REFINES

R12-4) กำหนดชื่อของแอ็บสแตร็คแมชชีนบี คือ Classi ในอนุประโยค REFINES

R12-5) กำหนดไลบรารีแอ็บสแตร็คแมชชีนบี และชื่อของแอ็บสแตร็คแมชชีนบี IntermediateClassi ในอนุประโยค SEES

R12-6) กำหนดอนุประโยค OPERATIONS

R12-7) ระบุโอเปอเรชัน Opi1() และชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ classi1 ในอนุประโยค OPERATIONS

R12-8) กำหนดอนุประโยค VARS

R12-9) ระบุชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ intermediateclassi1 ในอนุประโยค VARS

R12-10) กำหนดอนุประโยค IN

R12-11) ระบุโอเปอเรชัน Intermediate_Opi2() และชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ intermediateclassi1 ในอนุประโยค IN

R12-12) กำหนดอนุประโยค END เพื่อจบการทำงานของอนุประโยค OPERATIONS

R12-13) กำหนดอนุประโยค END เพื่อจบการทำงานของอิมพลีเทชันแอ็บสแตร็คแมชชีนบี Classi_imp

Opi2() เป็นโอเปอเรชันนอนเบซิคที่เรียกโอเปอเรชัน Opj() ภายใน Classi

R12-14) กำหนดอนุประโยค IMPLEMENTATION

R12-15) กำหนด IntermediateClassi_imp ในอนุประโยค IMPLEMENTATION

R12-16) กำหนดอนุประโยค REFINES

R12-17) กำหนดชื่อของแอ็บสแตร็คแมชชีนบี คือ IntermediateClassi ในอนุประโยค REFINES

R12-18) กำหนดไลบรารีแอ็บสแตร็คแมชชีนบี และชื่อของแอ็บสแตร็คแมชชีนบี BasicClassj ในอนุประโยค SEES

R12-19) กำหนดอนุประโยค OPERATIONS

R12-20) ระบุโอเปอเรชัน Opi2() และชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ intermediateclassi1 ในอนุประโยค OPERATIONS

R12-21) กำหนดอนุประโยค VARS

R12-22) ระบุชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ basicclassj1 ในอนุประโยค VARS

R12-23) กำหนดอนุประโยค IN

R12-24) ระบุโอเปอเรชัน Basic_Opj() และชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ basicclassj1 ในอนุประโยค IN

R12-25) กำหนดอนุประโยค END เพื่อจบการทำงานของอนุประโยค OPERATIONS

R12-26) กำหนดอนุประโยค END เพื่อจบการทำงานของอิมพลีเทชันแอ็บสแตร็คแมชชีนบี IntermediateClassi_imp

จากตัวอย่างรูปที่ 3.85 จะพบว่าโอเปอเรชัน getOrder(a1 : STRING) : STRING จะเรียกโอเปอเรชัน checkOrder(a1 : STRING) : BOOLEAN และโอเปอเรชัน checkOrder(a1 : STRING) : BOOLEAN เรียกโอเปอเรชัน getStock( ) : STRING ได้ข้อสรุปในการสร้างอิมพลิเมนต์เทชัน แอ็บสแตร็คแมชชีนบี ดังต่อไปนี้

1. โอเปอเรชัน getOrder( ) เป็นโอเปอเรชันนอนเบซิค ทำการระบุโอเปอเรชันนี้ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี Order โดยที่โอเปอเรชัน getOrder( ) เป็นโอเปอเรชันที่เรียกโอเปอเรชัน checkOrder( )

2. โอเปอเรชัน checkOrder( )เป็นโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชัน getOrder( ) ภายในคลาสเดิม ทำการสร้างแอ็บสแตร็คแมชชีนบี IntermediateOrder และทำการระบุโอเปอเรชันนี้ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี IntermediateOrder ใช้คำนำหน้า Intermediate แล้วตามด้วยโอเปอเรชัน getOrder( ) นั่นคือ Intermediate_getOrder()

3. โอเปอเรชัน getStock( ) เป็นโอเปอเรชันเบซิคที่ถูกเรียกจากโอเปอเรชันนอนเบซิค checkOrder( ) ทำการระบุโอเปอเรชันนี้ในอนุประโยค OPERATIONS ของแอ็บสแตร็คแมชชีนบี BasicStock โดยใช้คำนำหน้า Basic แล้วตามด้วย โอเปอเรชัน getStock( ) นั่นคือ Basic_getStock()

แสดงอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบี Order_imp และอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบี IntermediateOrder_imp ได้ดังรูปที่ 3.101 และรูปที่ 3.102

```
IMPLEMENTATION
Order_imp
REFINES
Order
SEES
...
IntermediateOrder
OPERATIONS
order1 ← getOrder(a1) =
VAR
boolean
IN
boolean ← Intermediate_checkOrder(a1)
END

END
```

รูปที่ 3.101 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี Order_imp

```
IMPLEMENTATION
IntermediateOrder_imp
REFINES
IntermediateOrder
SEES
...
BasicStock
OPERATIONS
boolean ← Intermediate_checkOrder(a1) =
VAR
basicstock1
IN
basicstock1 ← Basic_getStock
END

END
```

รูปที่ 3.102 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี IntermediateOrder_imp

กฎข้อที่ 13

การสร้างอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีเพื่อแสดงการเรียกโอเปอเรชันของซับคลาสที่มีคุณสมบัติของการถ่ายทอดโอเปอเรชันจากซูเปอร์คลาสจากเหตุการณ์ของแผนภาพซีเควนซ์

จากคุณสมบัติการถ่ายทอดโอเปอเรชันของซูเปอร์คลาส จะสามารถทำการสร้างอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบีของซับคลาสที่มีโอเปอเรชันที่เรียก นั่นคือ อิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบีของซับคลาสที่มีโอเปอเรชันที่เรียกจะแสดงการเรียกโอเปอเรชันของคลาสที่แสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คจากเหตุการณ์ของแผนภาพซีเควนซ์เช่นเดียวกับอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีของซูเปอร์คลาสที่มีโอเปอเรชันที่เรียก

จากรูปที่ 3.89 ซูเปอร์คลาสจะมีความสัมพันธ์กับ ClassX ในแผนภาพคลาส และมีการส่งข้อความระหว่างออบเจ็ค classX ของคลาส ClassX กับออบเจ็ค superclass ของคลาส SuperClass ในแผนภาพซีเควนซ์ จะพบว่าโอเปอเรชัน OpSup1 (ParamSup1 : TypeParamSup1) : TypeOpSup1 เรียกโอเปอเรชัน OpSup2 (ParamSup2 : TypeParamSup2) : TypeOpSup2

ดังนั้นอิมพลิเมนต์เทชันแอ็บสแตร์คแมชชีนบีของซับคลาส จะแสดงถึงโอเปอเรชัน OpSup1 (ParamSup1 : TypeParamSup1) : TypeOpSup1 เรียกโอเปอเรชัน OpSup2 (ParamSup2 : TypeParamSup2) : TypeOpSup2 เช่นเดียวกับอิมพลิเมนต์เทชันแอ็บสแตร์คแมชชีนบีของซูเปอร์คลาส แสดงได้ดังรูปที่ 3.103 และรูปที่ 3.104

```
IMPLEMENTATION
SuperClass_imp
REFINES
SuperClass
SEES
...
BasicSuperClass
OPERATIONS
superclass1 ← OpSup1(ParamSup1) =
VAR
basicsuperclass1
IN
basicsuperclass1 ← Basic_OpSup2(ParamSup2)
END

END
```

รูปที่ 3.103 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร์คแมชชีนบี SuperClass_imp

```
IMPLEMENTATION
SubClass1_imp
REFINES
SubClass1
SEES
...
BasicSubClass1
OPERATIONS
subclass11 ← OpSup1(ParamSup1) =
VAR
basicsubclass11
IN
basicsubclass11 ← Basic_OpSup2(ParamSup2)
END

END
```

```
IMPLEMENTATION
SubClass2_imp
REFINES
SubClass2
SEES
...
BasicSubClass2
OPERATIONS
subclass21 ← OpSup1(ParamSup1) =
VAR
basicsubclass21
IN
basicsubclass21 ← Basic_OpSup2(ParamSup2)
END

END
```

...

```
IMPLEMENTATION
SubClassN_imp
REFINES
SubClassN
SEES
...
BasicSubClassN
OPERATIONS
subclassN1 ← OpSup1(ParamSup1) =
VAR
basicsubclassN1
IN
basicsubclassN1 ← Basic_OpSup2(ParamSup2)
END

END
```

รูปที่ 3.104 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี SubClass1_imp จนถึง SubClassN_imp

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 13 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 13

สำหรับ SuperClassi ใด ๆ ในแผนภาพคลาส โดยที่ SuperClassi ใด ๆ จะถ่ายทอดโอเปอเรชันทั้งหมดนี้ให้กับ SubClassi ซึ่ง SubClassi จะมีโอเปอเรชันทั้งหมดที่มาจาก SuperClassi คือ OpSupi1() : TypeOpSupi1 และ OpSupi2() : TypeOpSupi2 พิจารณาในแผนภาพคลาสจะพบว่า ClassX มีความสัมพันธ์กับ SuperClass เมื่อพิจารณาเหตุการณ์ที่ k ในแผนภาพซีเควนซ์ ซึ่งออบเจ็ค classX ของคลาส ClassX จะมีการร้องขอให้ออบเจ็ค superclass ของคลาส SuperClass ทำโอเปอเรชัน OpSupi1() : TypeOpSupi1 ซึ่งออบเจ็ค superclass ของคลาส SuperClass จะทำ

โอเปอเรชัน OpSupi2() : TypeOpSupi2 โดยที่โอเปอเรชัน OpSupi1() จะเป็นโอเปอเรชันที่เรียกโอเปอเรชัน OpSupi2()

จากข้อความข้างต้นจะสามารถสรุปได้ว่า

1. โอเปอเรชัน OpSupi1() เป็นโอเปอเรชันนอนเบซิค
2. โอเปอเรชัน OpSupi2() เป็นโอเปอเรชันเบซิคที่เรียกภายใน Classi

จากข้อสรุปทั้ง 2 สามารถนำมาสร้างอิมพลีเทชันแอ็บสแตร็คแมชชีนบี SubClassi นั่นคือ SubClassi_imp ดังนี้

OpSupi1() เป็นโอเปอเรชันนอนเบซิคที่เรียกโอเปอเรชัน OpSupi2()

R13-1) กำหนดอนุประโยค IMPLEMENTATION

R13-2) กำหนด SubClassi_imp ในอนุประโยค IMPLEMENTATION

R13-3) กำหนดอนุประโยค REFINES

R13-4) กำหนดชื่อของแอ็บสแตร็คแมชชีนบี คือ SubClassi ในอนุประโยค REFINES

R13-5) กำหนดไลบรารีแอ็บสแตร็คแมชชีนบี และชื่อของแอ็บสแตร็คแมชชีนบี BasicSubClassi ในอนุประโยค SEES

R13-6) กำหนดอนุประโยค OPERATIONS

R13-7) ระบุโอเปอเรชัน OpSupi1() และชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ subclassi1 ในอนุประโยค OPERATIONS

R13-8) กำหนดอนุประโยค VARS

R13-9) ระบุชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ basicsubclassi1 ในอนุประโยค VARS

R13-10) กำหนดอนุประโยค IN

R13-11) ระบุโอเปอเรชัน Basic_OpSupi2() และชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ basicsubclassi1 ในอนุประโยค IN

R13-12) กำหนดอนุประโยค END เพื่อจบการทำงานของอนุประโยค OPERATIONS

R13-13) กำหนดอนุประโยค END เพื่อจบการทำงานของอิมพลีเทชันแอ็บสแตร็คแมชชีนบี SubClassi_imp

จากรูปที่ 3.94 แสดงตัวอย่าง คลาส Order เป็นซูเปอร์คลาสจะมีความสัมพันธ์แอสโซซิเอชันกับคลาส Customer และคลาส Stock รวมทั้งมีการส่งข้อความระหว่างออบเจ็คในแผนภาพซีเควนซ์ จะพบว่าโอเปอเรชัน getOrder(a1 : STRING) : STRING เรียกโอเปอเรชัน checkOrder(a1 :

STRING) : BOOLEAN และโอเปอเรชัน checkOrder(a1 : STRING) : BOOLEAN เรียกโอเปอเรชัน getStock( ) : STRING

ดังนั้นสร้างอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีของซับคลาส จะแสดงถึงโอเปอเรชัน getOrder ( ) จะเรียกโอเปอเรชัน checkOrder( ) และโอเปอเรชัน checkOrder( ) เรียกโอเปอเรชัน checkStock( ) เช่นเดียวกับอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบีของซูเปอร์คลาส

ทำการสร้างอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบี Goods_imp และทำการสร้างอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบี IntermediateGoods_imp โดยอาศัยคุณสมบัติของการถ่ายทอด โอเปอเรชันจากซูเปอร์คลาส แสดงได้ดังรูปที่ 3.105 ถึงรูปที่ 3.106

```
IMPLEMENTATION
Goods_imp
REFINES
Goods
SEES
...
IntermediateGoods
OPERATIONS
goods1 ← getOrder(a1) =
VAR
boolean
IN
boolean ← Intermediate_checkOrder(a1)
END

END
```

รูปที่ 3.105 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี Goods_imp

```
IMPLEMENTATION
IntermediateGoods _imp
REFINES
IntermediateGoods
SEES
...
BasicStock
OPERATIONS
boolean ← Intermediate_checkOrder(a1)
VAR
basicstock1
IN
basicstock1 ← Basic_getStock
END

END
```

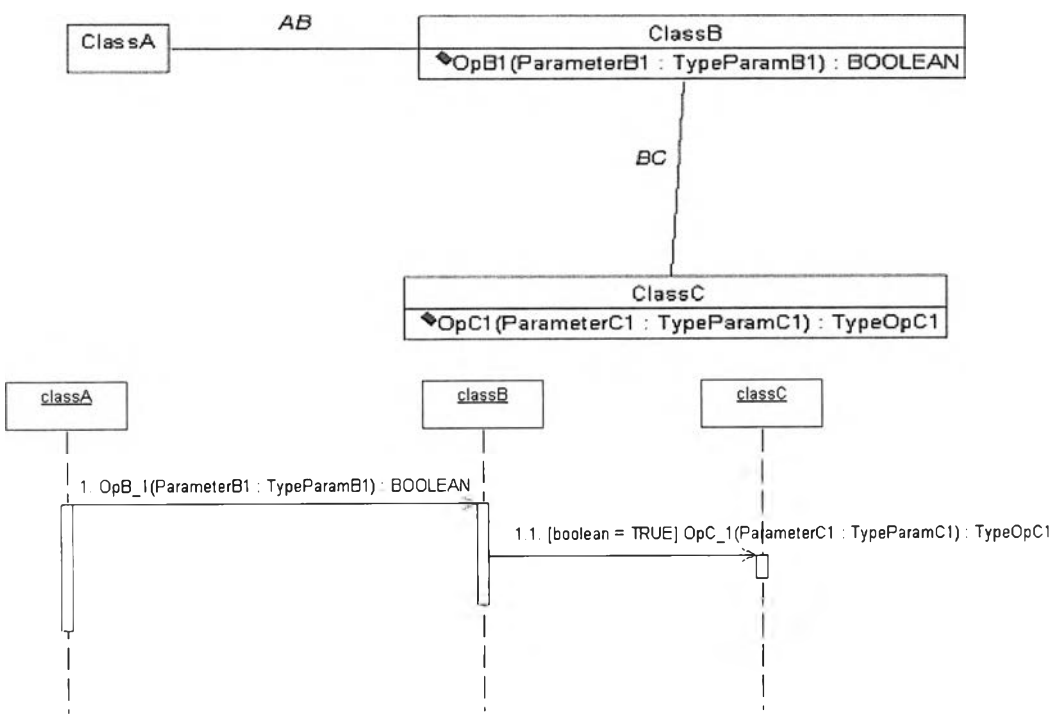
รูปที่ 3.106 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี IntermediateGoods_imp

กฎข้อที่ 14

การระบุเงื่อนไขที่เป็นจริงของโอเปอเรชันของคลาสที่แสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คจากเหตุการณ์ของแผนภาพซีเควนซ์ในอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี

ในกฎข้อนี้จะทำการระบุเงื่อนไขที่จะกระทำโอเปอเรชันของคลาสที่แสดงในรูปแบบการรับส่งข้อความระหว่างออบเจ็คที่แสดงจากเหตุการณ์ของแผนภาพซีเควนซ์ ในกรณีที่มีเงื่อนไขที่เป็นจริงของโอเปอเรชันที่เรียกจากเหตุการณ์ของแผนภาพซีเควนซ์ จะสามารถกระทำโอเปอเรชันที่ถูกเรียกนั้น ๆ ได้

พิจารณาแผนภาพคลาสที่แสดงความสัมพันธ์ระหว่าง ClassA กับ ClassB และความสัมพันธ์ระหว่าง ClassB กับ ClassC และแผนภาพซีเควนซ์ที่แสดงเหตุการณ์ของการรับส่งข้อความระหว่างออบเจ็ค classA ของคลาส ClassA และ classB ของคลาส ClassB และการรับส่งข้อความระหว่างออบเจ็ค classB ของคลาส ClassB และออบเจ็ค classC ของคลาส ClassC แสดงได้ดังรูปที่ 3.107



รูปที่ 3.107 รูปแบบของแผนภาพคลาสและแผนภาพซีเควนซ์ที่อธิบายในกฎข้อที่ 14

จากรูปที่ 107 จะพบว่า ClassB จะมีโอเปอเรชัน OpB1(Parameter1 : TypeParam1) : BOOLEAN และ ClassC จะมีโอเปอเรชัน OpC1(Parameter1 : TypeParam1) : TypeOpC1 และจากเหตุการณ์การรับส่งข้อความระหว่างออบเจ็ค classA ของคลาส ClassA กับออบเจ็ค classB ของคลาส ClassB และออบเจ็ค classB ของคลาส ClassB กับออบเจ็ค classC ของคลาส ClassC จะพบว่าถ้า OpB1( ) มีชนิดค่าที่ส่งคืน คือ ค่าบูลีนที่เป็นจริง จะสามารถสั่งให้ออบเจ็ค classC ของ

คลาส ClassC กระทำโอเปอเรชัน OpC1( ) ได้ ทั้งนี้จะพบว่าถ้ามีเงื่อนไขคือ [boolean = TRUE] สามารถทำโอเปอเรชัน OpC1( ) ตามที่ออบเจ็ค classB ของคลาส ClassB ต้องการได้ แสดงรูปแบบของการเรียกโอเปอเรชันจากเหตุการณ์ใน อิมพลีเมนเทชันแอ็บสแตร็คแมชชีนบี ClassB_imp ได้ดังรูปที่ 108

```
IMPLEMENTATION
ClassB_imp
REFINE
ClassB
SEES
BasicClassC
...
OPERATIONS
boolean ← OpB1(ParameterB1) =
VAR
basicclassc1
IN
        IF
        boolean = TRUE
        THEN
        basicclassc1 ← Basic_OpC1(ParameterC1)
        ELSE
        skip
        END
END

END
```

รูปที่ 3.108 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี ClassB_imp

สามารถพิจารณาขั้นตอนวิธีของกฎข้อที่ 14 ได้ดังนี้

ขั้นตอนวิธีของกฎข้อที่ 14

สำหรับ Classi Classj และ ClassX ในแผนภาพคลาส โดยที่ Classi จะมีโอเปอเรชัน Opi() : BOOLEAN ส่วน Classj จะมีโอเปอเรชัน Opj() : TypeOpj พิจารณาในแผนภาพคลาสจะพบว่า ClassX จะมีความสัมพันธ์ใด ๆ กับ Classi และ Classi จะมีความสัมพันธ์ใด ๆ กับ Classj จะสามารถแสดงได้ในเหตุการณ์ที่ k ของแผนภาพซีเควนซ์ นั่นคือ ออบเจ็ค classX ของคลาส ClassX จะมีการร้องขอให้ออบเจ็ค classi ของคลาส Classi ทำโอเปอเรชัน Opi() : BOOLEAN และออบเจ็ค classi ของคลาส Classi จะมีการร้องขอให้ออบเจ็ค classj ของคลาส Classj ทำโอเปอเรชัน [boolean = TRUE] Opj() ได้ก็ต่อเมื่อ เงื่อนไขของโอเปอเรชัน Opi() เป็นจริง โดยที่โอเปอเรชัน Opi() : BOOLEAN จะเป็นโอเปอเรชันที่เรียกโอเปอเรชัน Opj()จากข้อความข้างต้นจะสามารถสรุปได้ว่า

1. โอเปอเรชัน Opi() เป็นโอเปอเรชันนอนเบซิค
2. โอเปอเรชัน Opj() เป็นโอเปอเรชันเบซิคที่เรียกภายใน Classi

จากข้อสรุปทั้ง 2 สามารถนำมาสร้างอิมพลีเทชันแอ็บสแตร็คแมชชีนบี Classi นั่นคือ Classi_imp พร้อมทั้งสามารถระบุเงื่อนไขที่เป็นจริงของโอเปอเรชัน Opj() ในอิมพลีเทชันแอ็บสแตร็คแมชชีนบี Classi หรือ Classi_imp ได้ดังนี้

Opi1() **เป็นโอเปอเรชันนอนเบซิคที่เรียกโอเปอเรชัน** Opi2()

R14-1) กำหนดอนุประโยค IMPLEMENTATION

R14-2) กำหนด Classi_imp ในอนุประโยค IMPLEMENTATION

R14-3) กำหนดอนุประโยค REFINES

R14-4) กำหนดชื่อของแอ็บสแตร็คแมชชีนบี คือ Classi ในอนุประโยค REFINES

R14-5) กำหนดไลบรารีแอ็บสแตร็คแมชชีนบี และชื่อของแอ็บสแตร็คแมชชีนบี BasicClassi ในอนุประโยค SEES

R14-6) กำหนดอนุประโยค OPERATIONS

R14-7) ระบุโอเปอเรชัน Opi1() และชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ classi1 ในอนุประโยค OPERATIONS

R14-8) กำหนดอนุประโยค VARS

R14-9) ระบุชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ basicclassi1 ในอนุประโยค VARS

R14-10) กำหนดอนุประโยค IN

R14-11) กำหนดอนุประโยค IF

R14-12) กำหนดเงื่อนไขของตัวแปร Boolean = TRUE ในอนุประโยค IF

R14-13) กำหนดอนุประโยค THEN

R14-14) ระบุโอเปอเรชัน Basic_Opj() และชนิดค่าที่ส่งคืนของโอเปอเรชันในรูปตัวแปรผลลัพธ์คือ basicclassi1 ในอนุประโยค THEN

R14-15) กำหนดอนุประโยค ELSE

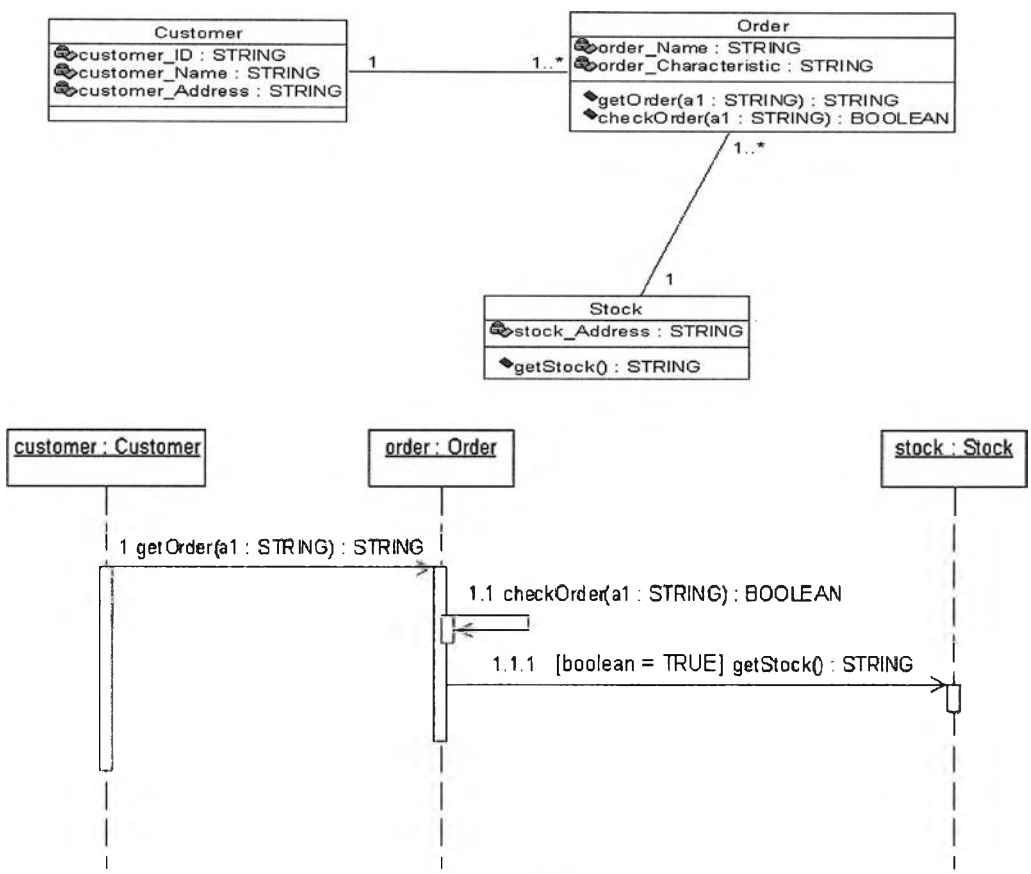
R14-16) กำหนดอนุประโยค skip

R14-17) กำหนดอนุประโยค END เพื่อจบการทำงานในส่วนของเงื่อนไข IF...THEN...ELSE...

R14-18) กำหนดอนุประโยค END เพื่อจบการทำงานของอนุประโยค OPERATIONS

R14-19) กำหนดอนุประโยค END เพื่อจบการทำงานของอิมพลีเทชันแอ็บสแตร็คแมชชีนบี Classi_imp

จากตัวอย่างของแผนภาพคลาสที่แสดงส่วนหนึ่งของระบบการสั่งซื้อสินค้า และตัวอย่างของแผนภาพซีเควนซ์ที่แสดงเหตุการณ์การสั่งซื้อสินค้าจากลูกค้า เมื่อพิจารณาแผนภาพคลาสจะพบว่าคลาส Order มีโอเปอเรชัน getOrder(a1 : STRING) : STRING และโอเปอเรชัน checkOrder (a1 : STRING) : BOOLEAN ส่วนคลาส Stock มีโอเปอเรชัน getStock() : STRING โดยพิจารณาจากแผนภาพซีเควนซ์ ของเหตุการณ์การสั่งซื้อสินค้าจากลูกค้า จะพบว่ามีการส่งข้อความระหว่างออบเจ็ค customer ของคลาส Customer กับออบเจ็ค order ของคลาส Order และออบเจ็ค order ของคลาส Order กับออบเจ็ค stock ของคลาส Stock แสดงได้ดังรูปที่ 3.109



รูปที่ 3.109 ตัวอย่างแผนภาพคลาสและซีเควนซ์ระบบการสั่งซื้อสินค้าโดยระบุเงื่อนไขของ getStock()

จากรูปที่ 3.109 จะพบว่าในแผนภาพคลาส และจากแผนภาพซีเควนซ์ของระบบการสั่งซื้อสินค้า จะพบว่าโอเปอเรชัน getOrder(a1 : STRING) : STRING เรียกโอเปอเรชัน checkOrder (a1 : STRING) : BOOLEAN และโอเปอเรชัน checkOrder(a1 : STRING) : BOOLEAN จะสามารถเรียกโอเปอเรชัน getStock( ) : STRING ให้ทำการ getStock( ) ได้ก็ต่อเมื่อชนิดค่าที่ส่งคืนจากโอเปอเรชัน checkOrder() มีค่าบูลีนเป็นจริง นั่นคือ boolean = TRUE ทำให้ได้ข้อสรุปเพื่อสร้างอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบี ดังต่อไปนี้

1. โอเปอเรชัน getOrder( ) เป็นโอเปอเรชันนอนเบซิค ทำการระบุโอเปอเรชันนี้ในแอ็บสแตร็คแมชชีนบี Order โดยที่โอเปอเรชัน getOrder( ) เป็นโอเปอเรชันที่เรียกโอเปอเรชัน checkOrder( )
2. โอเปอเรชัน checkOrder( )เป็นโอเปอเรชันนอนเบซิคที่ถูกเรียกจากโอเปอเรชัน getOrder( ) ภายในคลาสเดิม ทำการสร้างแอ็บสแตร็คแมชชีนบี IntermediateOrder และทำการระบุโอเปอเรชันนี้ในแอ็บสแตร็คแมชชีนบี IntermediateOrder ใช้คำนำหน้า Intermediate แล้วตามด้วยโอเปอเรชัน checkOrder( ) นั่นคือ Intermediate_checkOrder()
3. โอเปอเรชัน getStock( ) เป็นโอเปอเรชันเบซิค ทำการระบุโอเปอเรชันนี้ในแอ็บสแตร็คแมชชีนบี BasicStock ใช้คำนำหน้า Basic แล้วตามด้วยโอเปอเรชัน getStock( ) นั่นคือ Basic_getStock()

ทั้งนี้จะพบว่าถ้ามีเงื่อนไขคือ [boolean = TRUE] สามารถทำโอเปอเรชัน getStock( ) ตามที่ออบเจ็ค order ของคลาส Order ต้องการได้

แสดงอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบี Order_imp และอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบี IntermediateOrder_imp โดยการระบุเงื่อนไขที่เป็นจริงของโอเปอเรชัน Intermediate_CheckOrder() ในอิมพลิเมนต์เทชันแอ็บสแตร็คแมชชีนบี IntermediateOrder_imp ได้ดังรูปที่ 3.110 และรูปที่ 3.111

```
IMPLEMENTATION
Order_imp
REFINES
Order
SEES
...
IntermediateOrder
OPERATIONS
order1 ← getOrder(a1) =
VAR
boolean
IN
boolean ← Intermediate_checkOrder(a1)
END

END
```

รูปที่ 3.110 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี Order_imp

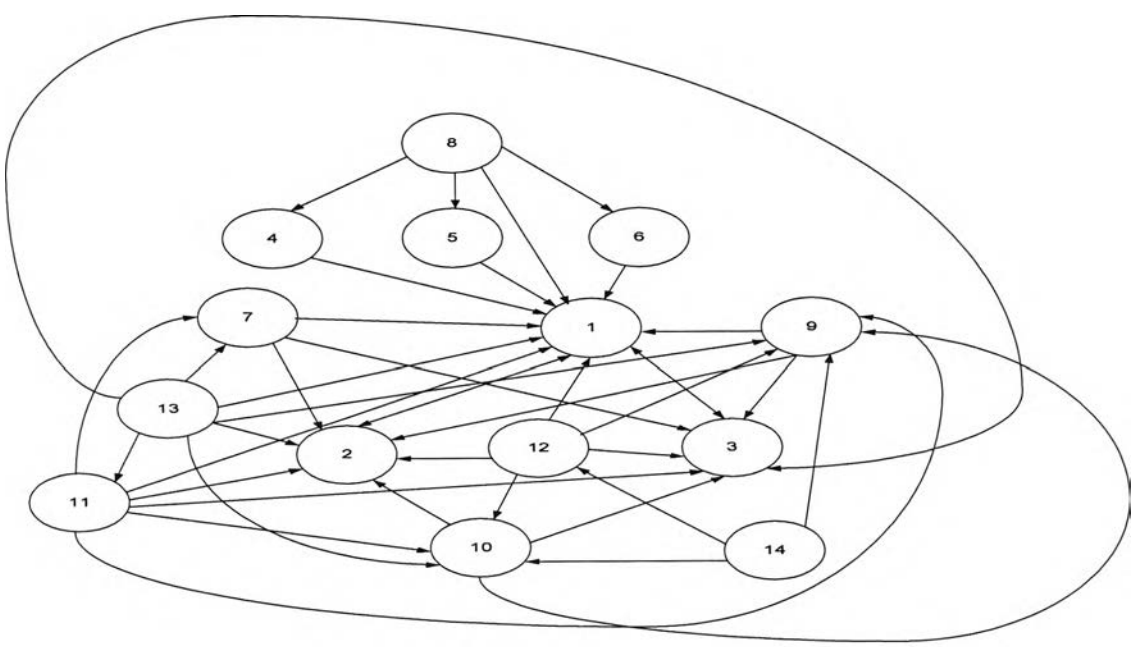
```
IMPLEMENTATION
IntermediateOrder_imp
REFINES
IntermediateOrder
SEES
...
Stock
OPERATIONS
boolean ← Intermediate_checkOrder(a1) =
VAR
basicstock1
IN
        IF
        boolean = TRUE
        THEN
        basicstock1 ← Basic_getStock
        ELSE
        skip
        END

END

END
```

รูปที่ 3.111 รูปแบบอิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบี IntermediateOrder_imp

จากกฎการแปลงแผนภาพคลาส และแผนภาพซีเควนซ์เป็นแอ็บสแตร็คแมชชีนบีทั้ง 14 ข้อ จะสามารถแสดงรูปแบบการขึ้นต่อกันของกฎทั้ง 14 ข้อได้ จะอาศัยแผนภาพการขึ้นต่อกัน (Dependency Diagram) เพื่อแสดงรูปแบบของการขึ้นต่อกันของกฎทั้ง 14 ข้อ โดยเครื่องหมายลูกศรแทนการขึ้นต่อกันของกฎ นั่นคือ กฎที่อยู่ทางปลายลูกศรจะขึ้นกับกฎที่อยู่ทางหัวลูกศร ดังรูปที่ 112



รูปที่ 3.112 แผนภาพการขึ้นต่อกันของกฎทั้ง 14 ข้อ

จากรูปที่ 112 ซึ่งแสดงแผนภาพการขึ้นต่อกันของกฎการแปลงทั้ง 14 ข้อ จะสามารถสร้างตารางการขึ้นต่อกัน (Dependency Matrix) ของกฎการแปลงทั้ง 14 ข้อ เพื่อให้ง่ายต่อการพิจารณารูปแบบของการขึ้นต่อกันของกฎทั้ง 14 ข้อ แสดงรูปแบบของตารางการขึ้นต่อกันของกฎการแปลงทั้ง 14 ข้อ ได้ดังตารางที่ 3.1

ตารางที่ 3.1 ตารางแสดงรูปแบบการขึ้นต่อกันของกฎการแปลงทั้ง 14 ข้อ

Dependency	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8	Rule 9	Rul 10	Rule 11	Rule 12	Rule 13	Rule 14
Rule-1		✓	✓											
Rule-2	✓													
Rule-3	✓													
Rule-4	✓													
Rule-5	✓													
Rule-6	✓													
Rule-7	✓	✓	✓											
Rule-8	✓			✓	✓	✓								
Rule-9	✓	✓	✓											
Rule-10		✓	✓						✓					
Rule-11	✓	✓	✓				✓		✓	✓				
Rule-12	✓	✓	✓						✓	✓				
Rule-13	✓	✓	✓				✓		✓	✓	✓			
Rule-14									✓	✓		✓		