

บทที่ 8

Activity Diagram

ในการพัฒนาโปรแกรม flowchart จัดเป็นเครื่องมือที่มีประสิทธิภาพ สามารถแสดงให้เห็นถึงลำดับขั้นตอนของการทำงาน , อธิบายกระบวนการทำงานต่าง ๆ รวมทั้งแสดงการตัดสินใจของระบบงานด้วยคุณสมบัตินี้ ภาษา UML จึงนำมารวมเป็นกลไกหนึ่งของภาษา โดยมีการปรับเปลี่ยนเล็กน้อยเพื่อความเหมาะสมแก่การใช้งานยิ่งขึ้นจนกลายเป็น “Activity Diagram”

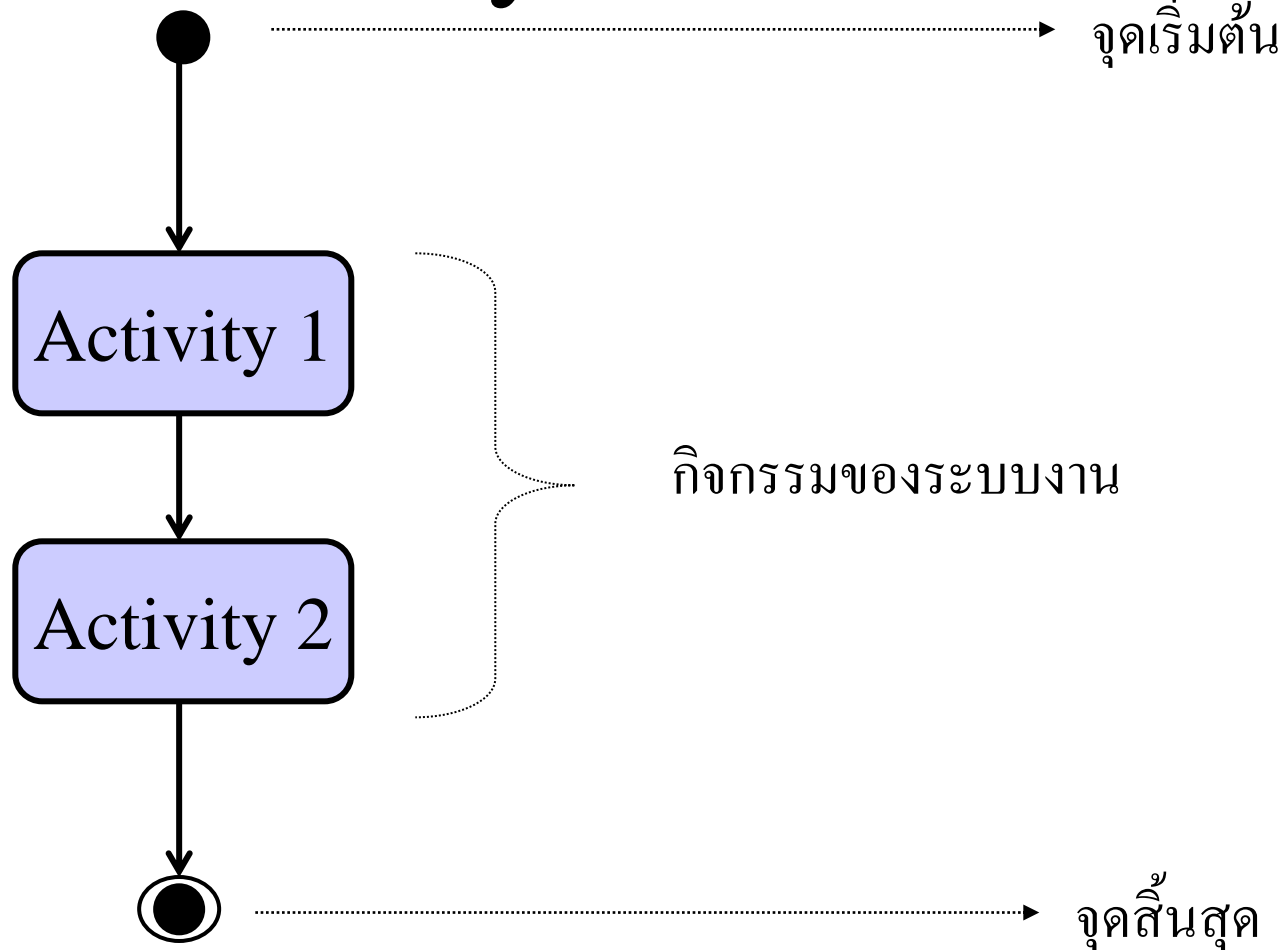
ลักษณะทั่วไปของ Activity diagram

จะมีลักษณะเดียวกับ flowchart คือใช้สำหรับแสดงขั้นตอนการทำงานของระบบ โดยมีขั้นตอนในการทำงานแต่ละขั้นตอนซึ่งเรียกว่า activity

Activity อาจเป็นลักษณะการทำงานต่าง ๆ ได้แก่

- การคำนวณผลลัพธ์บางอย่าง
- การเปลี่ยนแปลงสถานะ (state) ของระบบ
- การส่งค่าบางอย่างกลับคืนมา (return value)
- การเรียกให้ method อื่น ๆ ทำงาน
- การส่งสัญญาณ
- การสร้าง หรือการทำงาน object

activity diagram จะต้องมีจุดเริ่มต้น และ
จุดสิ้นสุด และในระหว่างจุดเริ่มต้นกับจุดสิ้นสุด ก็จะมีขั้นตอนหรือ activity ต่าง ๆ ของระบบ ดังรูป



รูปแบบการใช้ activity diagram มีหลายแบบ ได้แก่

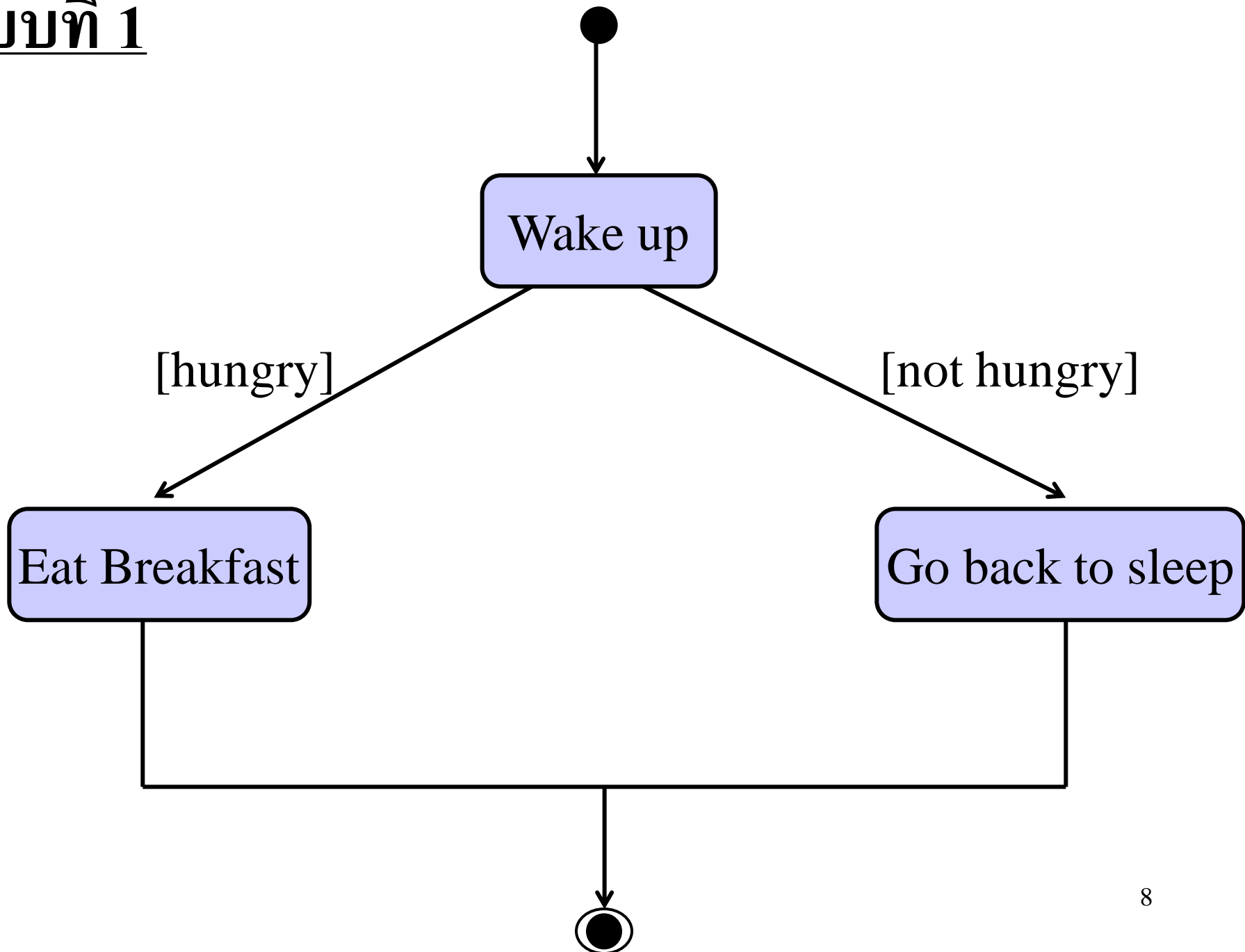
1. แบบทั่วไป
2. แบบมีทางเลือกให้ตัดสินใจ
3. แบบที่มีการทำงานพร้อม ๆ กันหลายงาน
4. Activity diagram สำหรับการส่งสัญญาณ

การสร้างทางเลือกให้แก่ activity diagram

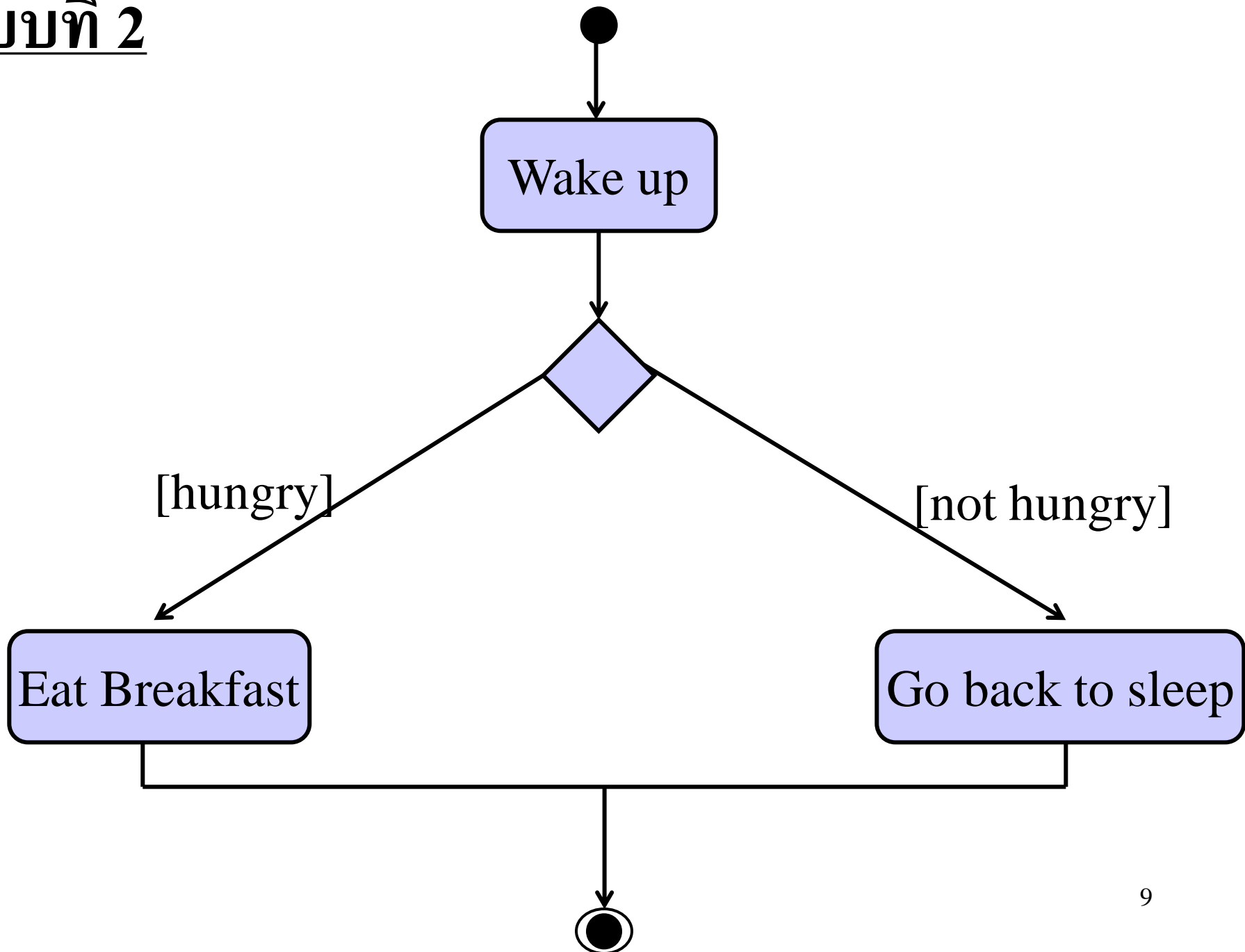
สามารถทำได้ 2 วิธีคือ

1. ลากลูกศรของแต่ละทางเลือกไปยัง activity
ผลลัพธ์ของทางเลือกโดยตรง
2. ลากลูกศรให้ลูกศรของแต่ละทางเลือกผ่านรูปสี่เหลี่ยมขนมเปียกปูนก่อน (เหมือน flowchart)

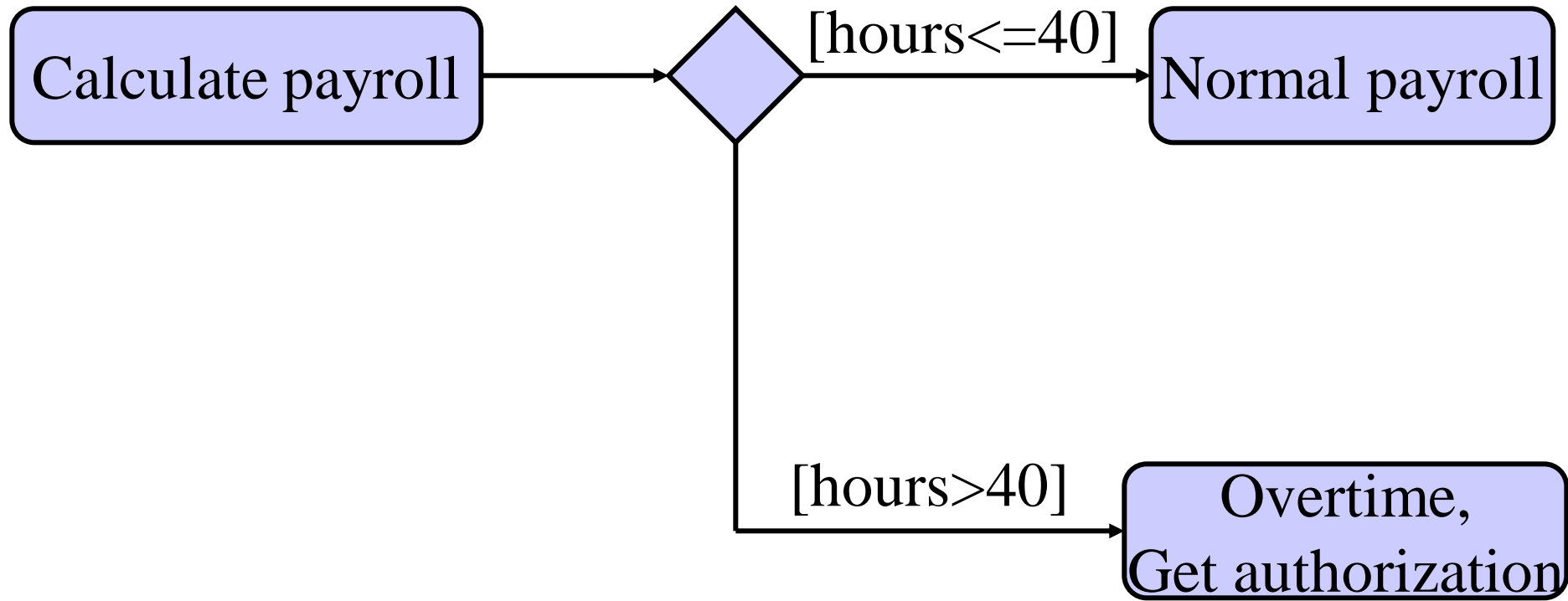
แบบที่ 1



แบบที่ 2

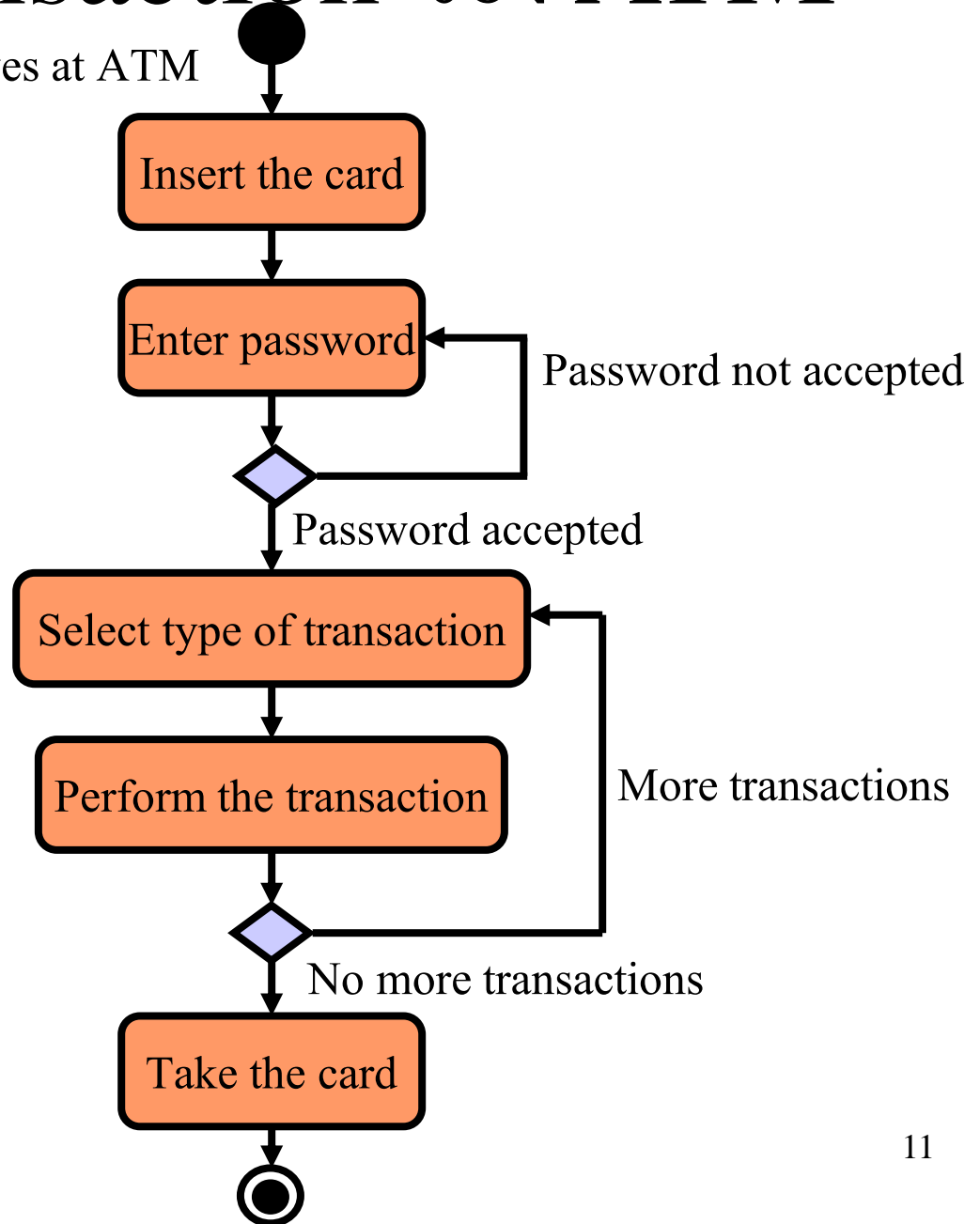


ตัวอย่าง การตัดสินใจ

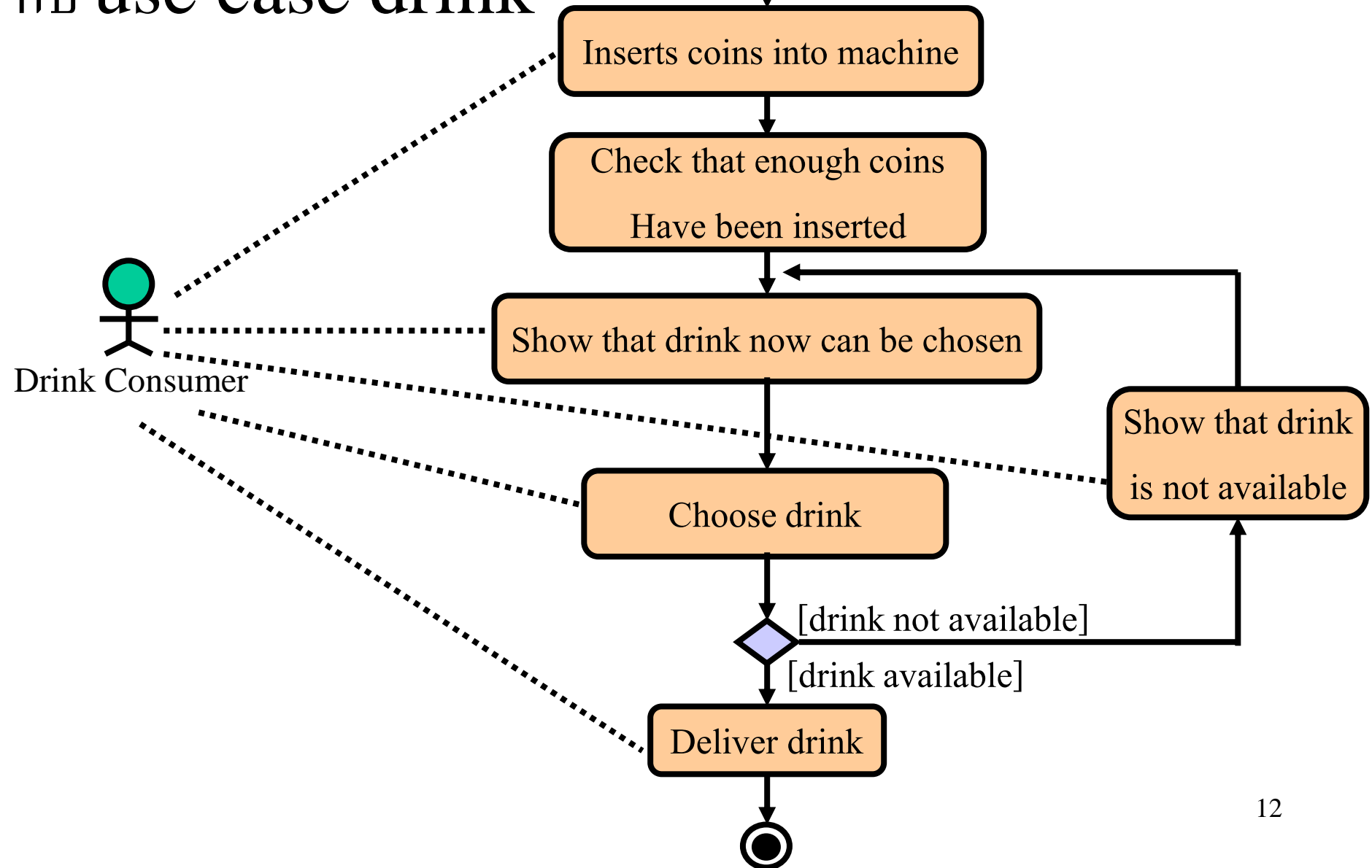


ตัวอย่าง การทำ transaction ของ ATM

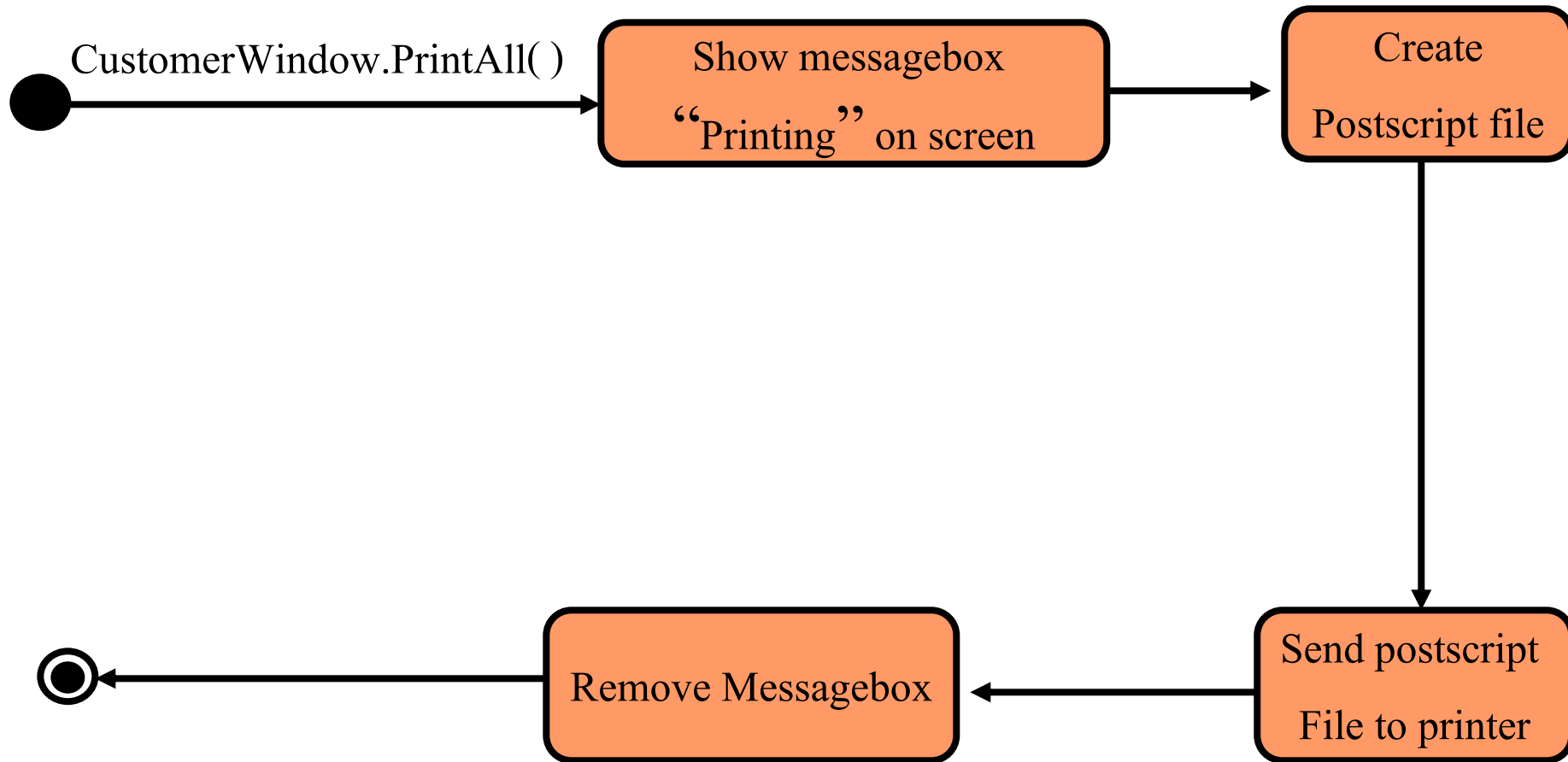
Member arrives at ATM



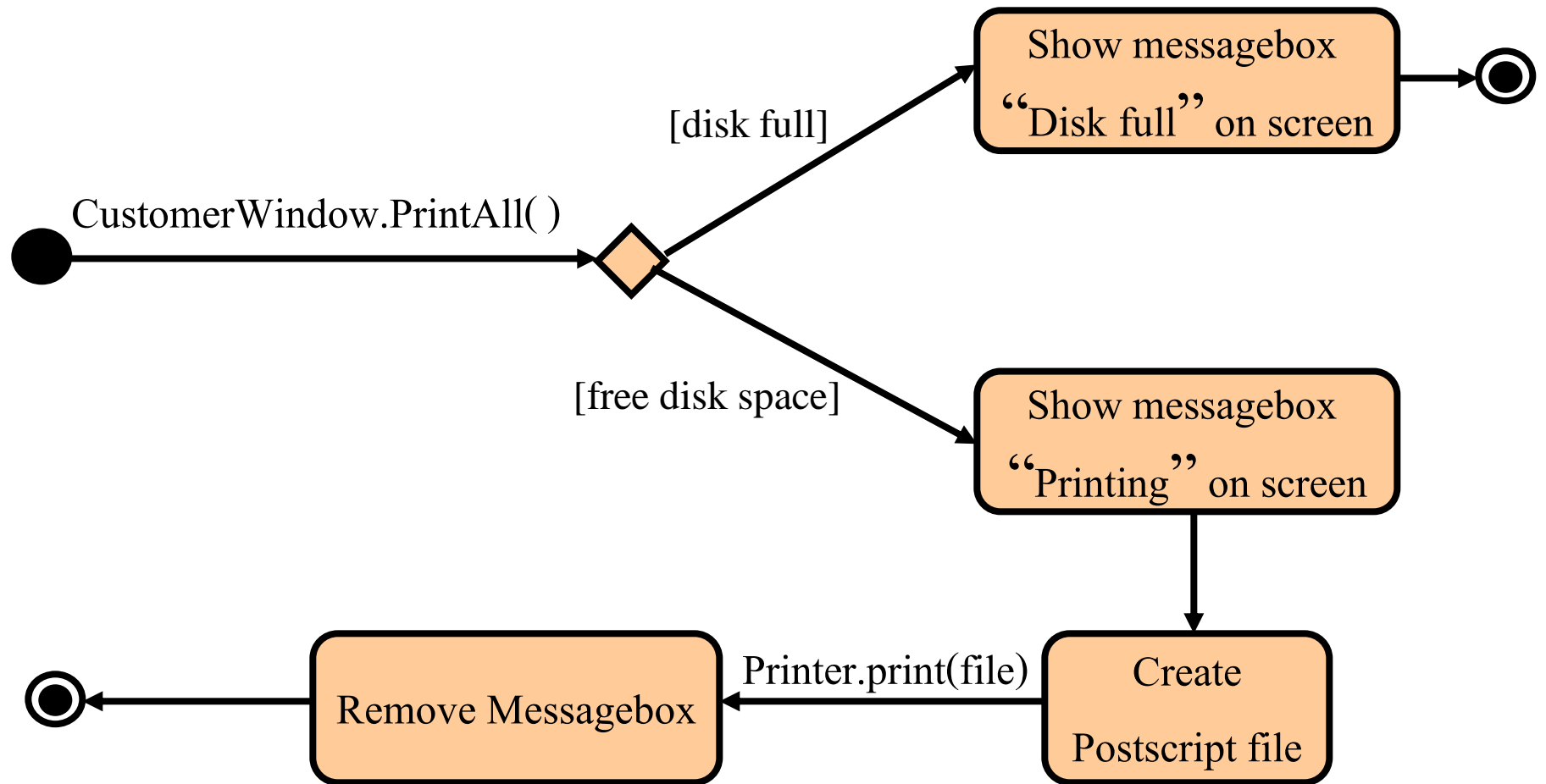
ตัวอย่าง การทำงานของเครื่องขายน้ำอัดลม ซึ่งมาจาก actor ติดต่อกับ use case drink



ตัวอย่าง การทำงานของ print server

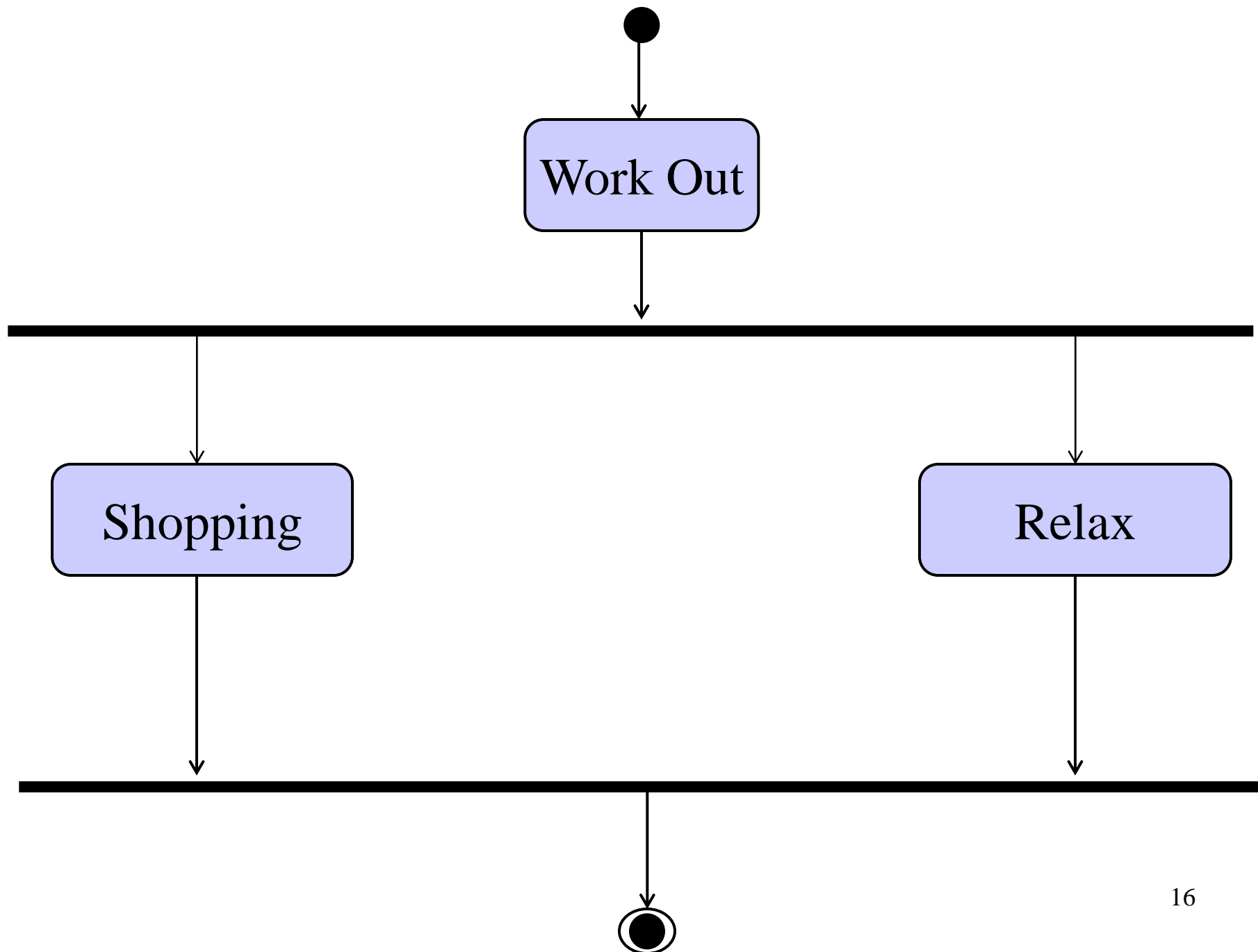


ตัวอย่าง การทำงานของ print server แบบมีการ ตรวจสอบขนาดของ server

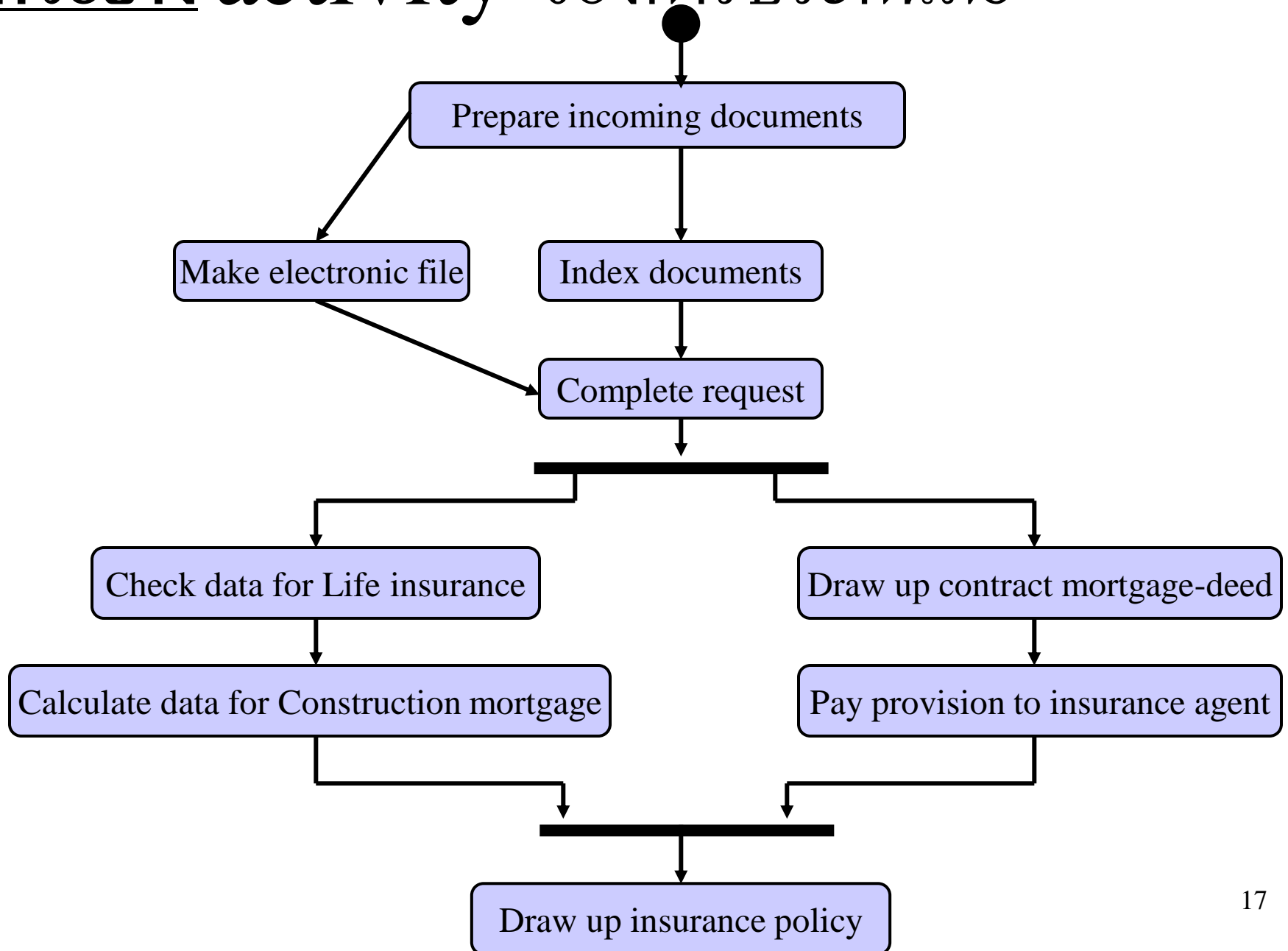


การทำงานไปพร้อมกันหลาย ๆ งาน

ในกรณีที่เรามีงานหลายงานที่มีการทำงานไปพร้อมกัน จะใช้เส้นตรงแนวนอนเส้นหนา มาเป็นสัญลักษณ์ที่ใช้จัดกลุ่มงานที่มีการทำงานไปพร้อม ๆ กัน โดยมีลักษณะดังรูป



ตัวอย่าง activity ของการประกันภัย



Activity Diagram แสดงการส่งสัญญาณ

ในกระบวนการทำงานอาจเป็นไปได้ว่าจะมีการส่งสัญญาณบางอย่างในระหว่างการทำงาน เมื่อเกิดการส่ง-รับสัญญาณ เราก็จะเรียกว่าเกิด **activity** ขึ้นเช่นเดียวกัน

ในการเขียน activity diagram สำหรับ
การส่งสัญญาณ จะใช้รูปหลายเหลี่ยมแทน activity
ที่มีการส่งสัญญาณ โดยที่

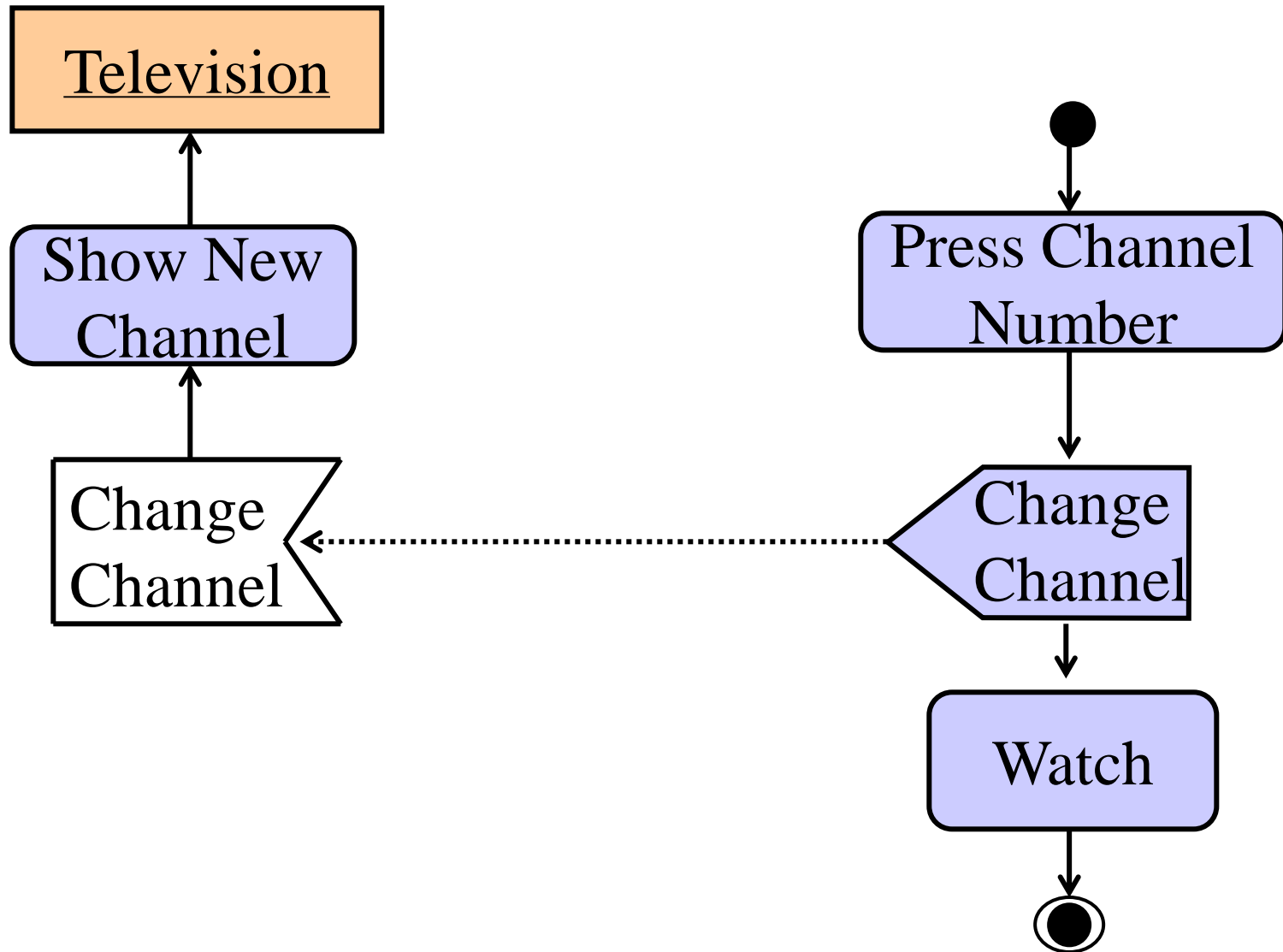


แทนเหตุการณ์ที่เป็น input

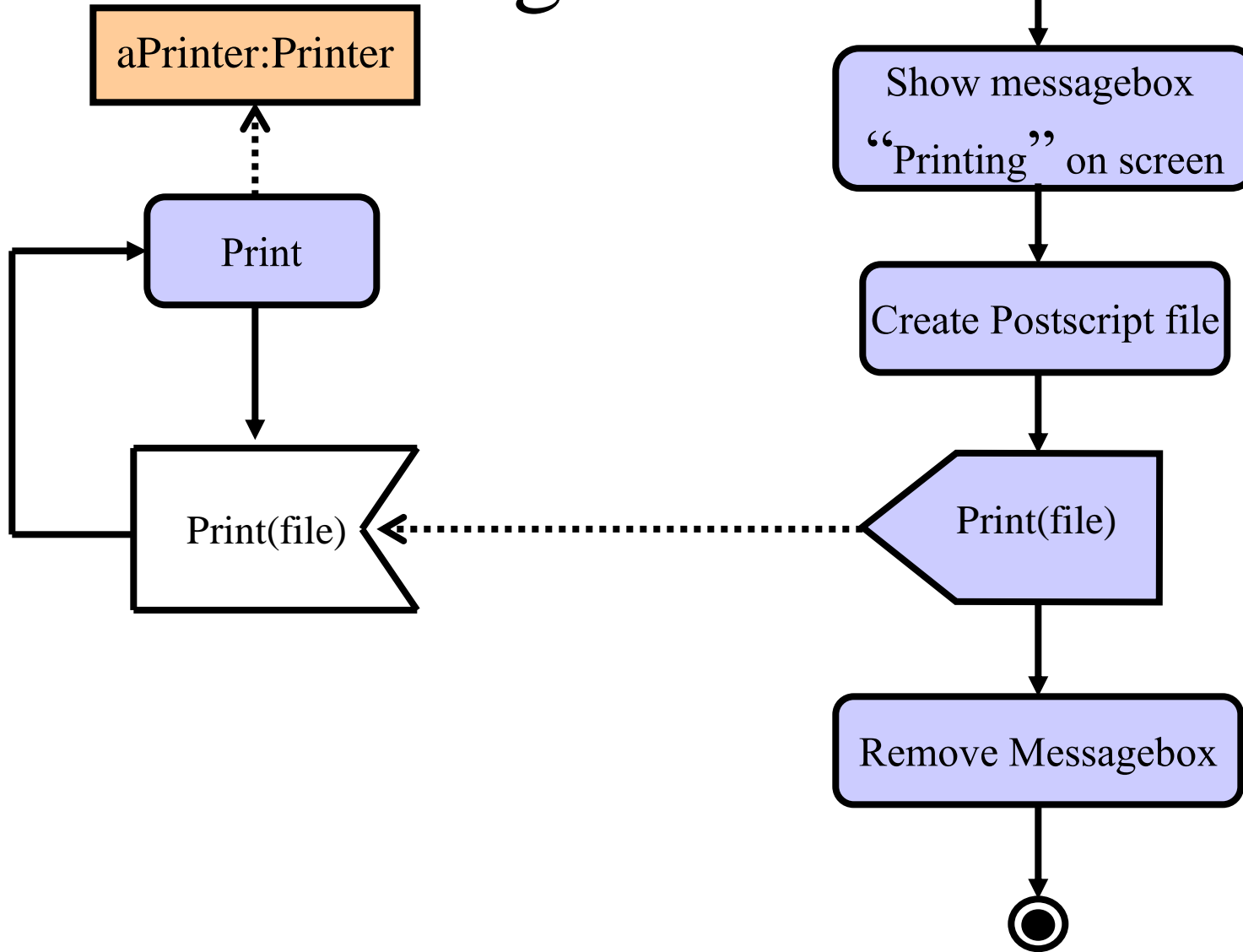


แทนเหตุการณ์ที่เป็น output

ตัวอย่าง แสดงการส่งสัญญาณ โดยระบบที่สนใจคือ
การกดปุ่มรีโมทคอนโทรลเพื่อเปลี่ยนช่องโทรทัศน์



ตัวอย่าง แสดงการส่งสัญญาณ ในช่วงการสร้างการพิมพ์งาน และการลบ messagebox



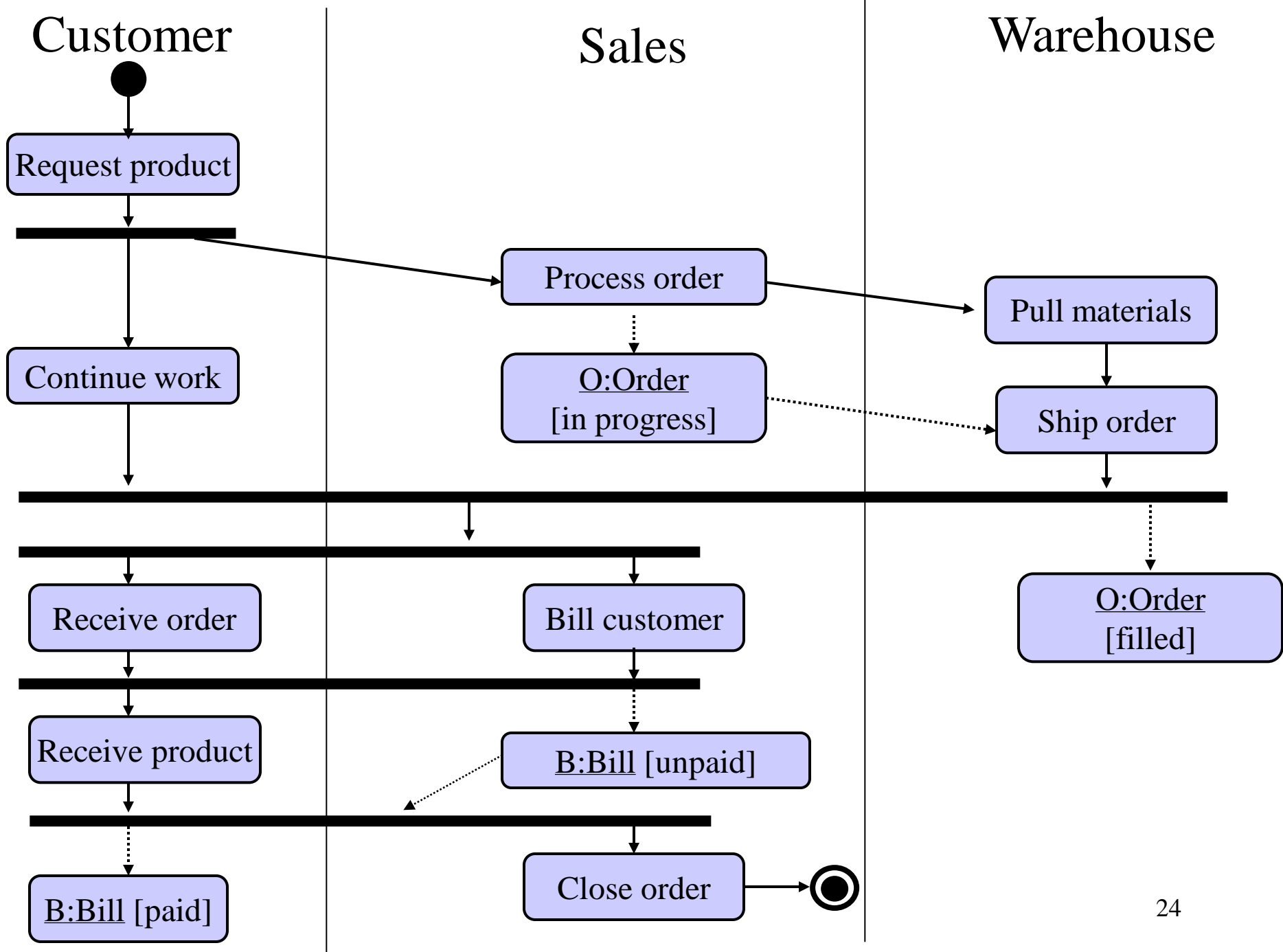
แบ่งการทำงานให้เป็นสัดส่วนด้วย Swimlanes

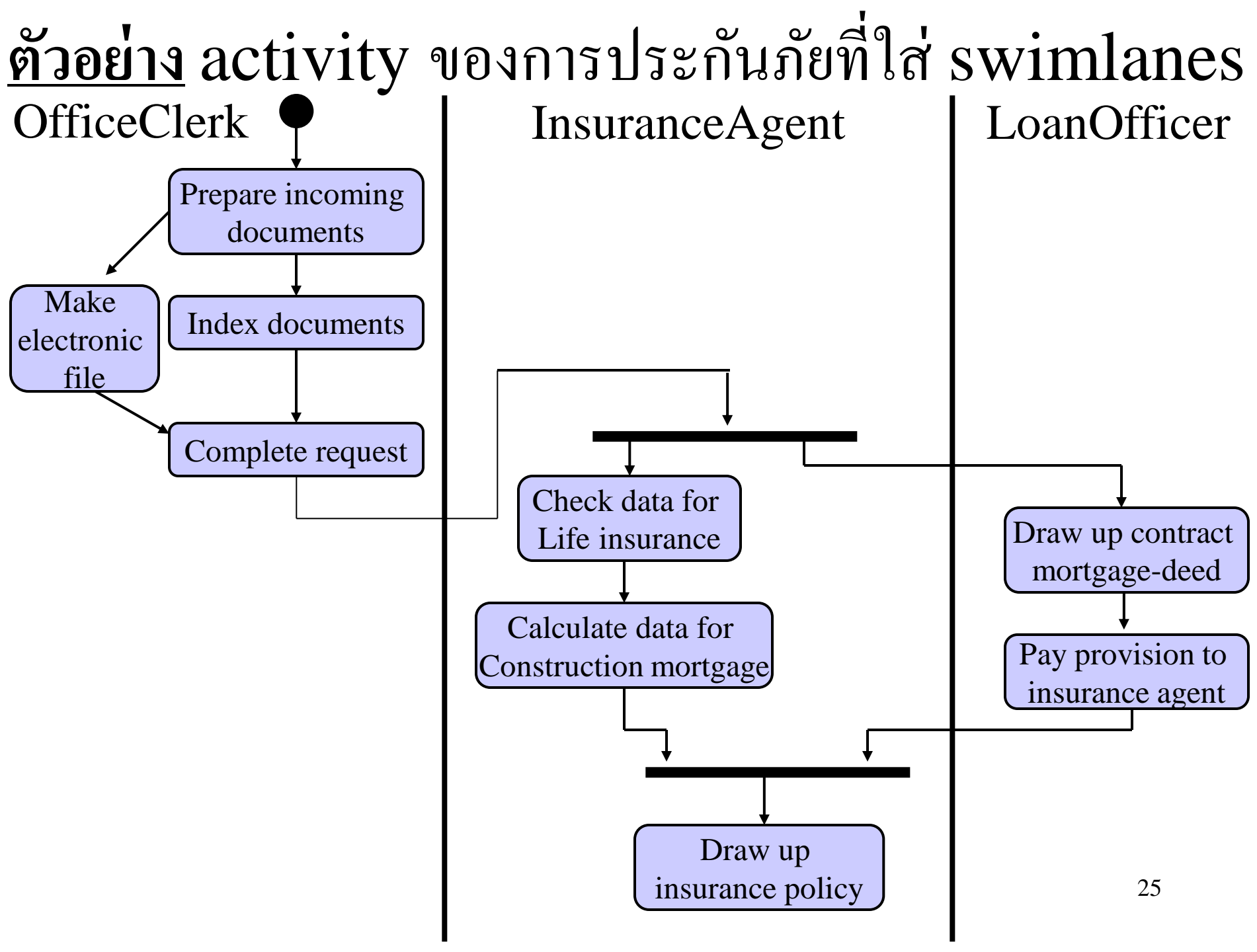
คุณลักษณะอีกอย่างหนึ่งคือสามารถแสดงให้เห็นได้ว่าใครเป็นผู้มีหน้าที่รับผิดชอบในแต่ละ activity ในกระบวนการทำงานหนึ่ง ๆ

หลักการของการแสดงหน้าที่ จะทำโดยการแบ่งกลุ่มของการรับผิดชอบเป็นกลุ่ม ๆ ซึ่งเปรียบเหมือนการแข่งขันว่ายน้ำ เรียกกลไกนี้ว่า “Swimlanes”

ในแต่ละ swimlane จะมีการกำหนดชื่อกำกับเอาไว้ เช่นกระบวนการของการสั่งซื้อสินค้า เราอาจแบ่งกลุ่มของคนที่มีส่วนเกี่ยวข้องเป็น 3 ส่วน ได้แก่ ลูกค้า , ฝ่ายขาย และคลังสินค้า

activity หนึ่ง ๆ จะอยู่ภายใน 1 swimlane เท่านั้น แต่การติดต่อหรือส่งผ่านระหว่าง activity สามารถเกิดขึ้นข้ามจาก swimlane หนึ่งไปยังอีก swimlane หนึ่งได้ ดังรูป





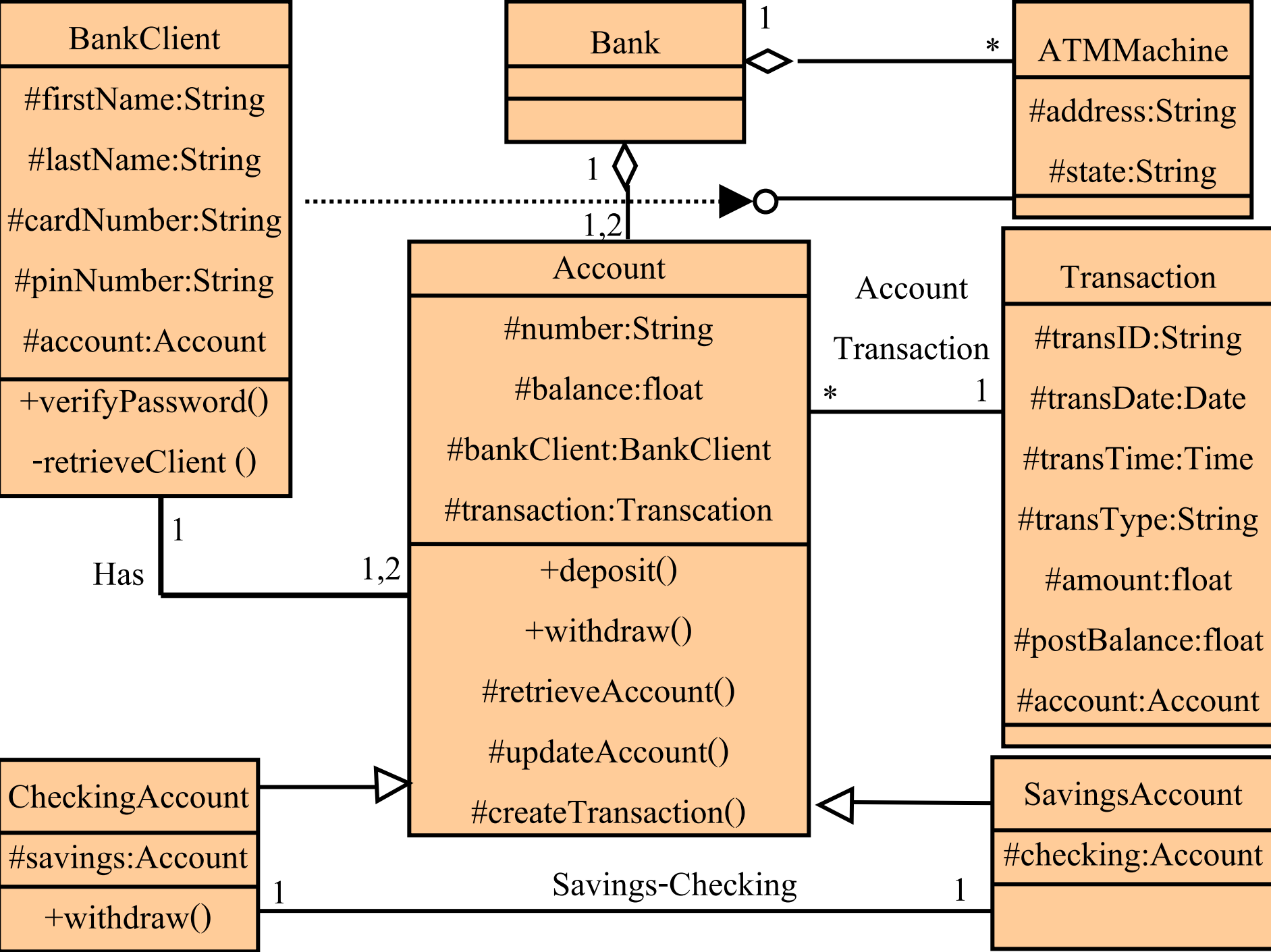
ข้อแตกต่างของ activity diagram และ state diagram อยู่ที่ลักษณะของการเปลี่ยนแปลงของระบบ ถ้าเป็นการเปลี่ยนแปลงหรือเหตุการณ์ต่าง ๆ อันเป็นผลจากการกระทำภายในระบบที่มีลักษณะต่อเนื่องกัน และมีจุดที่ต้องตัดสินใจ ให้ใช้ activity diagram แต่ถ้าการเปลี่ยนแปลงหรือการกระทำที่เกิดขึ้นในลักษณะไม่ต่อเนื่องกัน ให้ใช้ state diagram

- Activity diagram ที่ดีควรมีคุณสมบัติต่อไปนี้
- มุ่งเน้นไปที่การติดต่อสื่อสารของระบบในเชิง **dynamic** คือเป็นการติดต่อที่สภาพระบบมีการเปลี่ยนแปลงไม่หยุดนิ่ง
 - เฉพาะ **element** ที่มีความสำคัญต่อกระบวนการทำงานเท่านั้น
 - เลือกลงรายละเอียดในแต่ละระดับการทำงาน เฉพาะที่มีความสำคัญต่อการเข้าใจการทำงานของระบบเท่านั้น

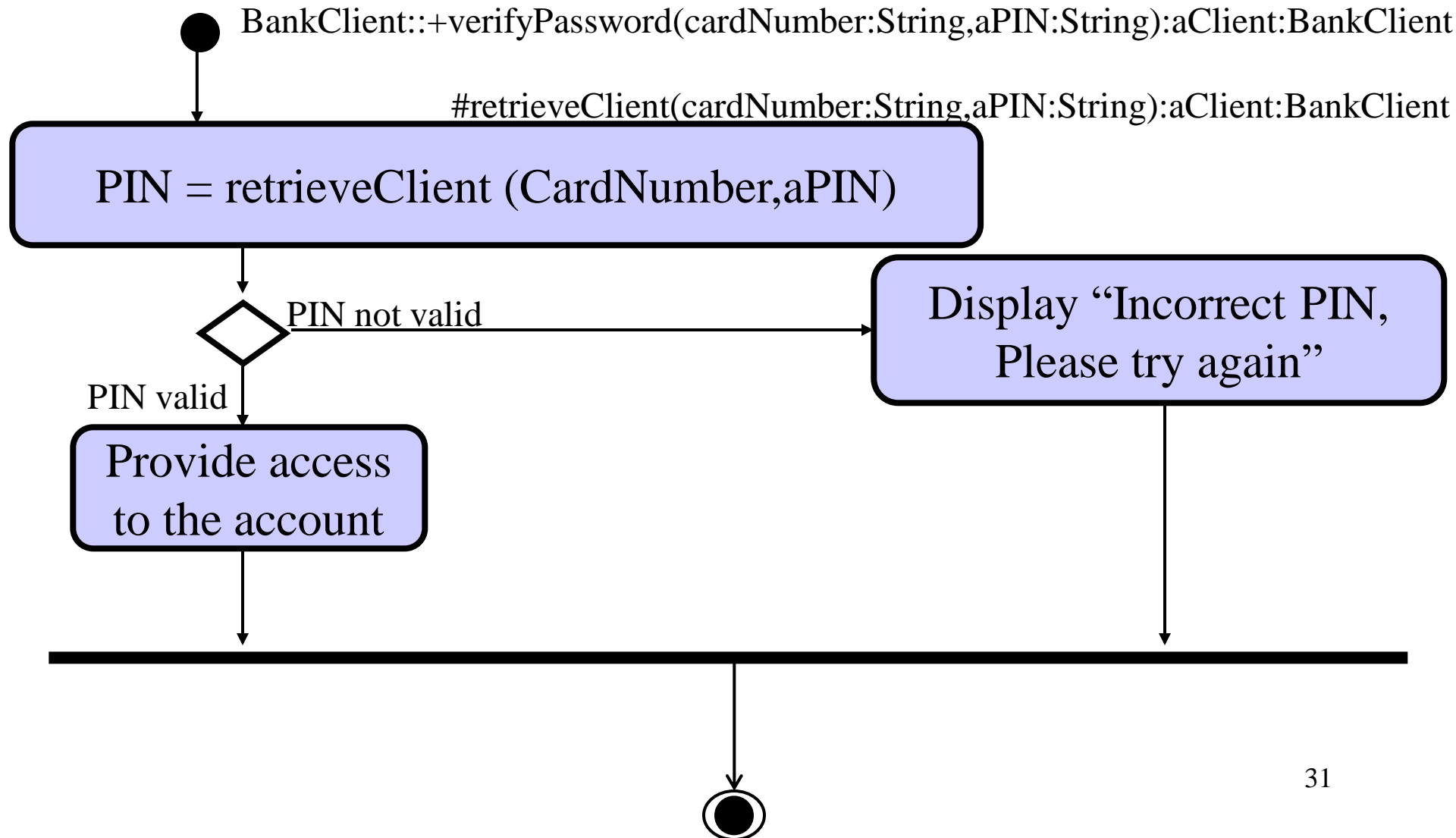
- สรุปวิธีการสร้าง Activity diagram มี 4 ข้อ คือ
1. ตั้งชื่อให้แก่ activity diagram ที่สร้าง และบอกเป้าหมายของการสร้างในแต่ละโคอะแกรมด้วย
 2. ให้เริ่มต้นสร้างโฟลว์การทำงานหลัก ๆ ก่อนแล้วจึงลงรายละเอียดในส่วนของการตัดสินใจ , การทำงานหลายอย่างพร้อม ๆ กัน และรายละเอียดย่อย
 3. จัดวาง element ต่าง ๆ ให้อ่านเข้าใจได้ง่าย
 4. อาจใช้ note ช่วยให้โคอะแกรมอ่านได้ง่ายขึ้นหรือเน้นในส่วนรายละเอียดที่ต้องการให้ความสำคัญ

Case study (ต่อ)

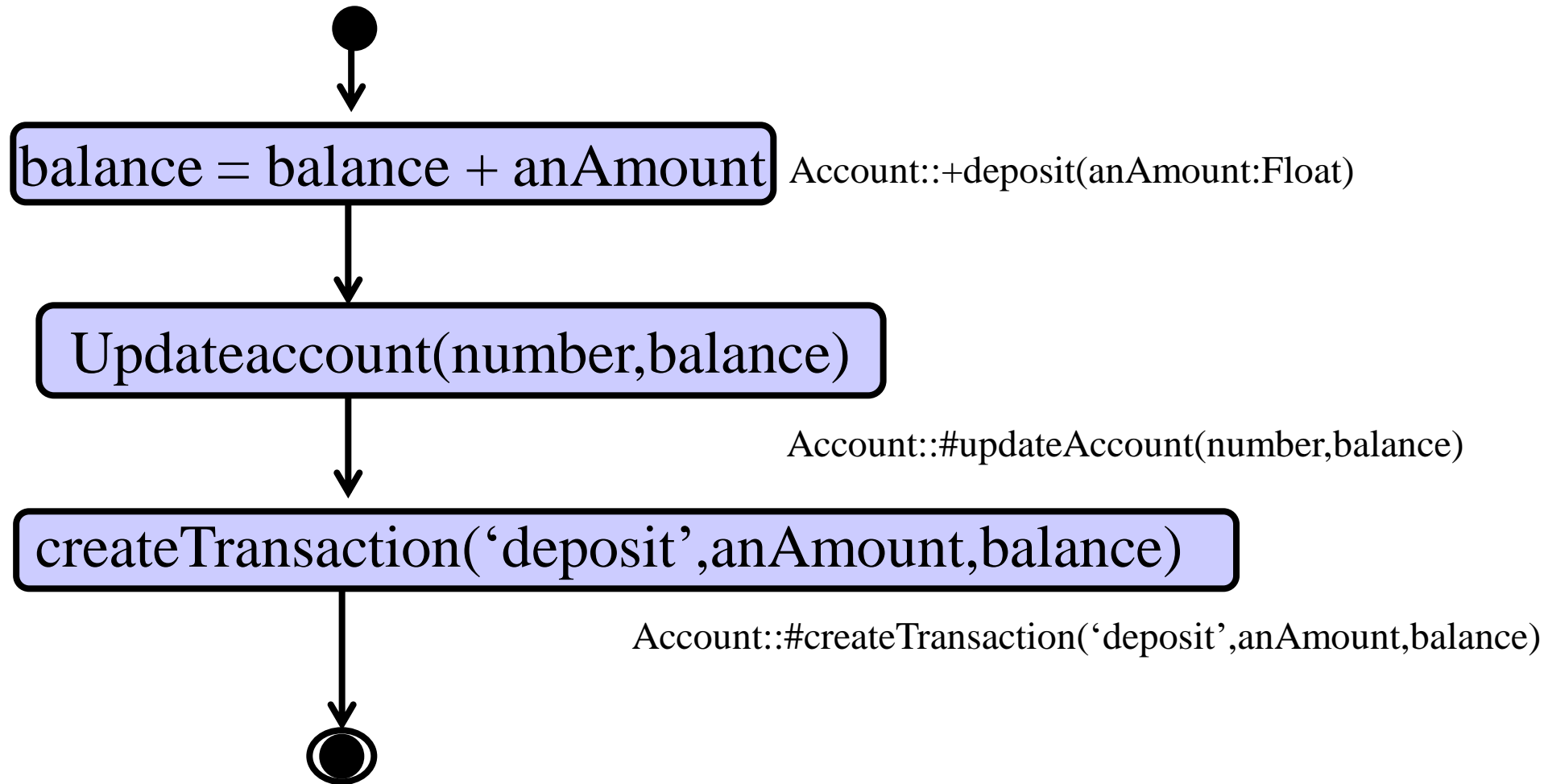
จาก class diagram ที่ได้เมื่อนำมาวิเคราะห์
หา attribute และ method ที่สำคัญจะได้ดังนี้



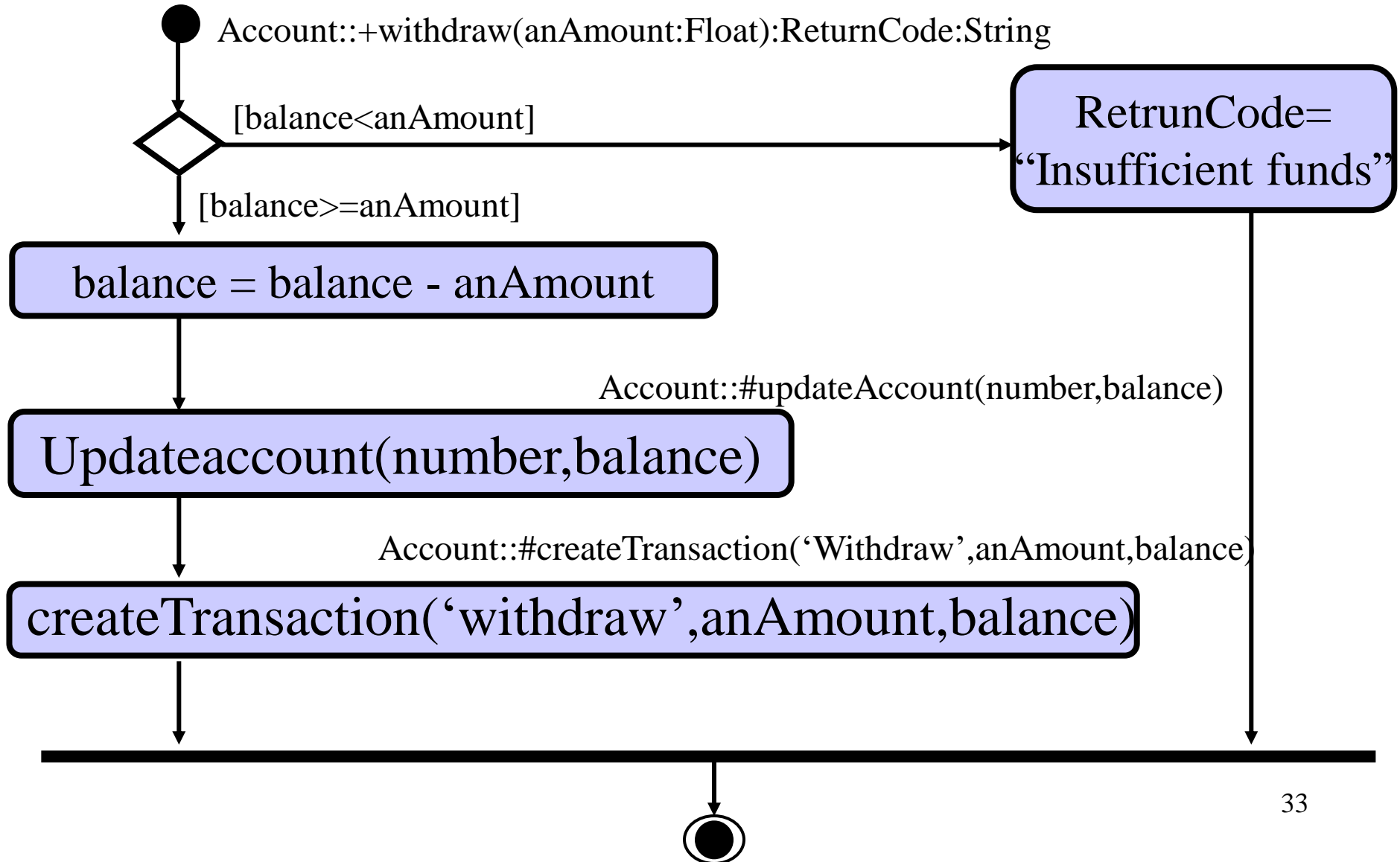
ตัวอย่าง activity diagram จาก method verifyPassword ของ class BankClient



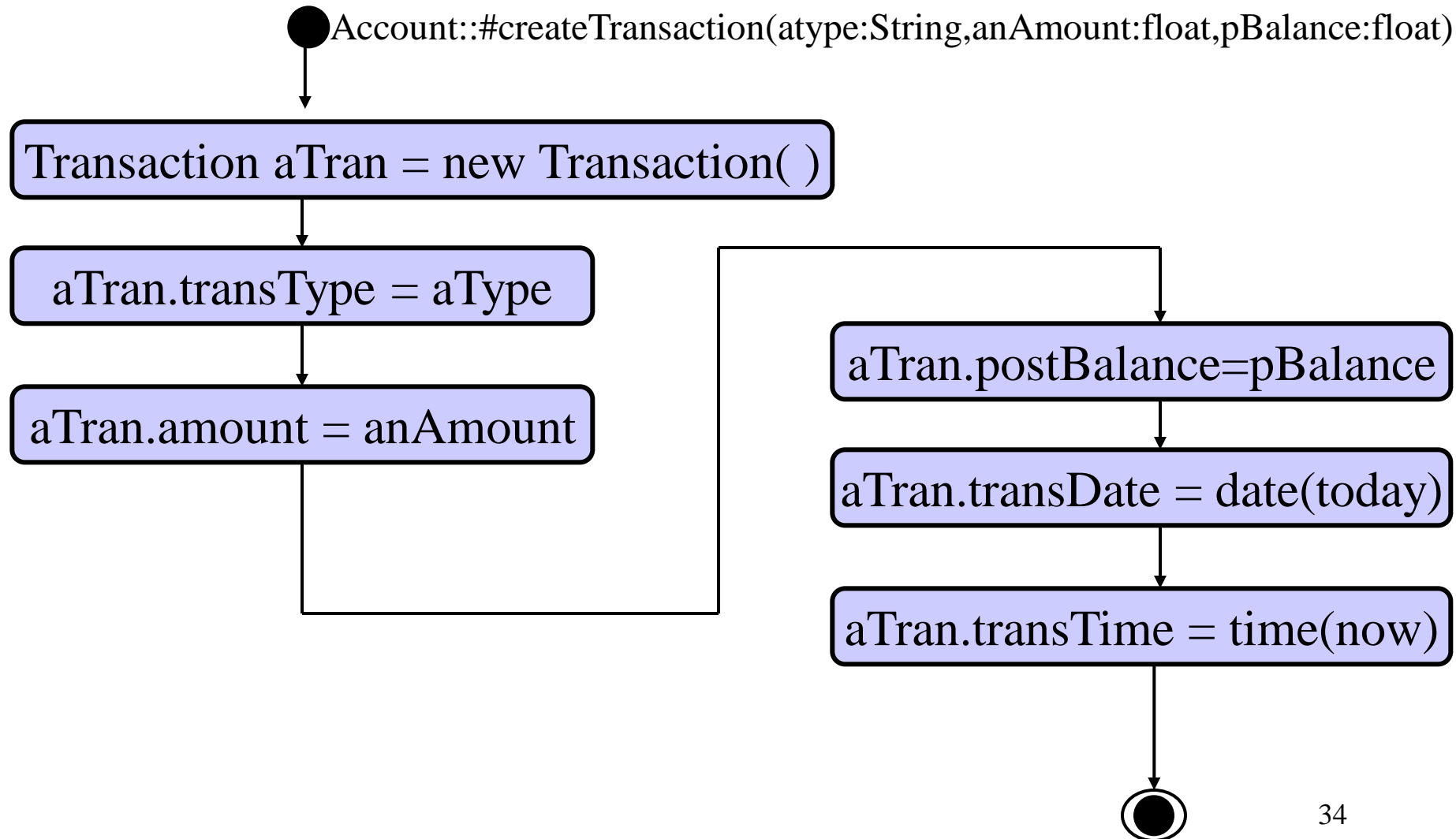
ตัวอย่าง activity diagram จาก method deposit ของ class Account



ตัวอย่าง activity diagram จาก method withdraw ของ class Account



ตัวอย่าง activity diagram จาก method createTransaction ของ class Account



ตัวอย่าง activity diagram จาก method withdraw ของ class CheckingAccount

