L7: State Machine Diagrams

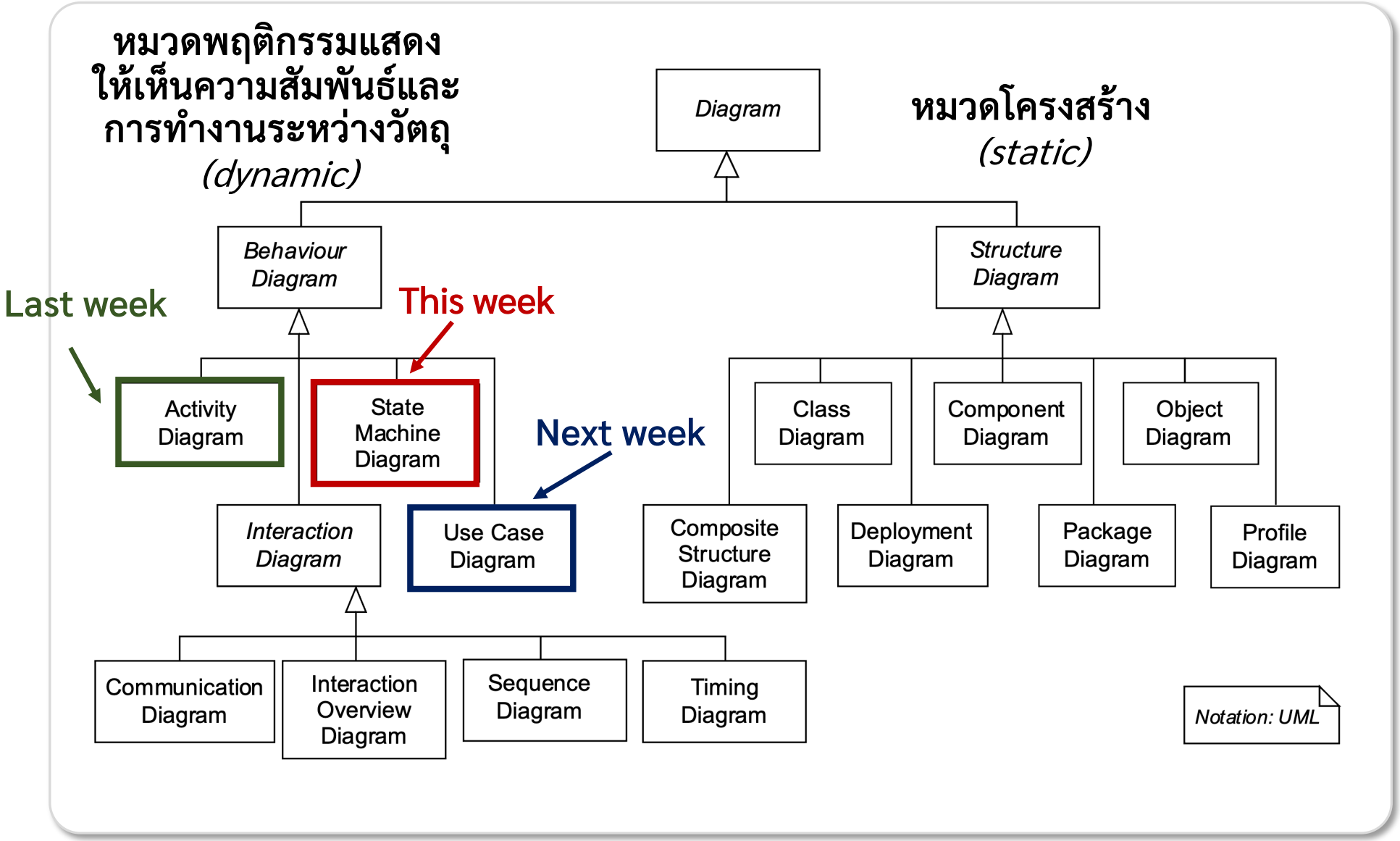
Kamonphop Srisopha

Faculty of Science, Chiang Mai University
คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่

Agenda

- State Machine Diagram and its notations
- Activity Diagrams vs State Machine Diagrams

UML Diagrams



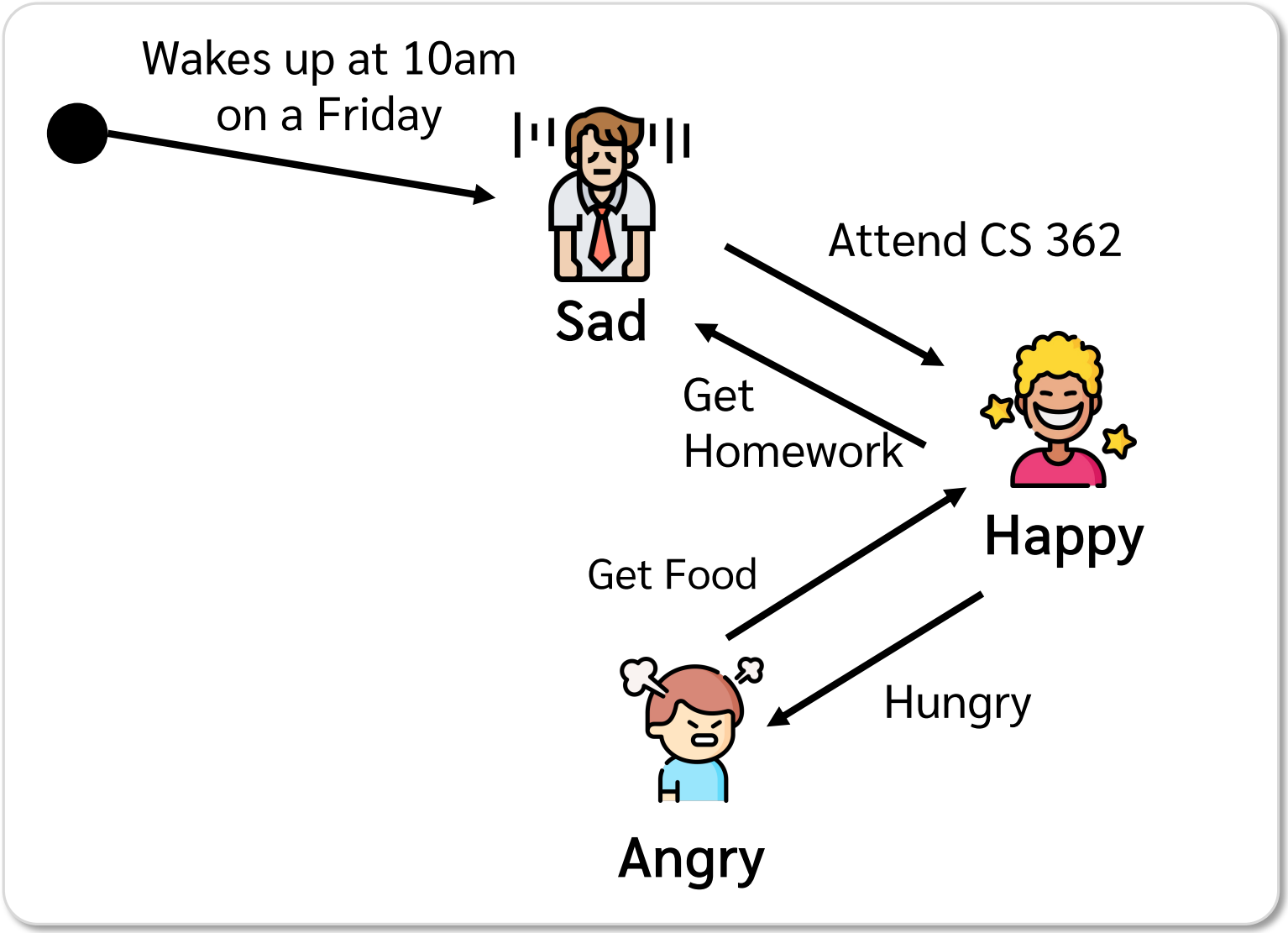
State Machine Diagrams

State Machine Diagrams

- เป็นแผนภาพส่วนหนึ่งของ Behavior Diagram ที่ใช้แสดงพฤติกรรมต่างๆของระบบ (หรือ object ของคลาสใดคลาสหนึ่งที่ซับซ้อน)
- มีอีกหลายชื่อเรียกเช่น State transition diagram, Statechart, State diagram เป็นต้น

เน้นการแสดงแผนภาพของ สถานะ (State)
และการเปลี่ยนแปลงของสถานะ (Transition)
ต่อเหตุการณ์ (Event) ต่างๆที่เกิดขึ้น

States of a ‘Student’ Class Object for CS 362



What is a State?

- ณ เวลาใดเวลาหนึ่ง ระบบจะต้องอยู่ในสถานะ *(state)* หนึ่ง ขึ้นอยู่กับค่าของลักษณะ *(attributes)* ของมันในตอนนั้น
- ระบบจะคงอยู่ในสภานะนั้นจนกว่าจะเกิดเหตุการณ์ *(event)* ที่ทำให้มันเปลี่ยนสถานะ *(transition)*
- การอยู่ในสถานะหมายถึง**ระบบมีพฤติกรรมจำเพาะเพื่อตอบสนองต่อเหตุการณ์บางอย่างที่เกิดขึ้น**
- บางเหตุการณ์จะทำให้ระบบเปลี่ยนสถานะ และ **เมื่อมันอยู่ในสถานะใหม่ มันจะมีพฤติกรรมที่แตกต่างออกไปตอบสนองต่อเหตุการ์ณแตกต่างออกไปจากสถานะก่อน**

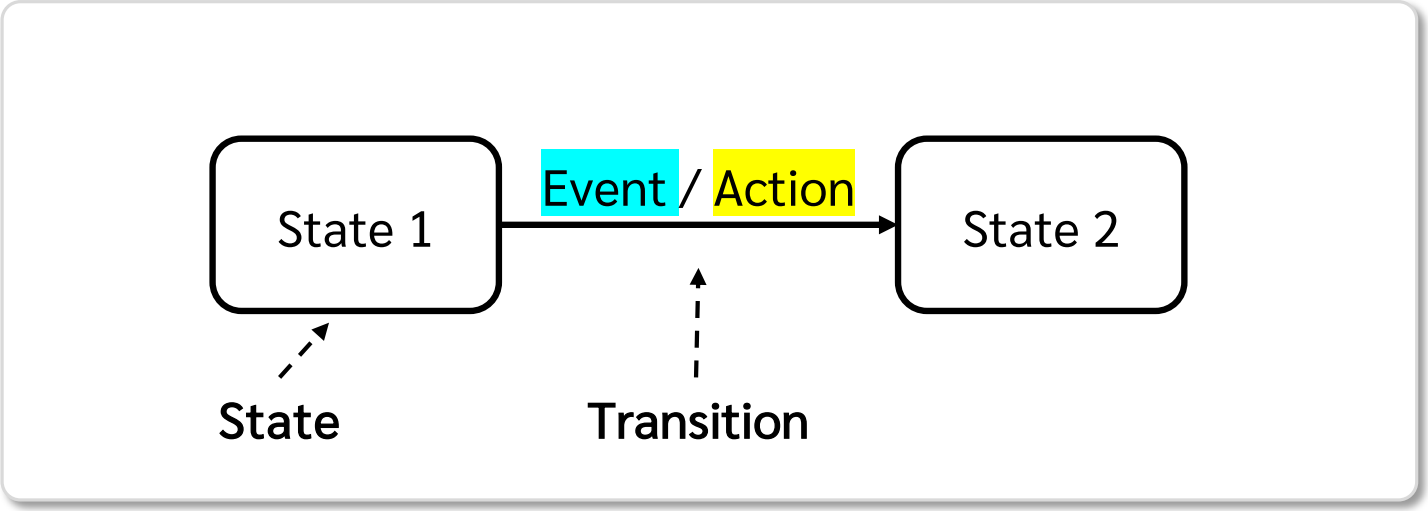
States of Things

Overview of State Machine Diagrams

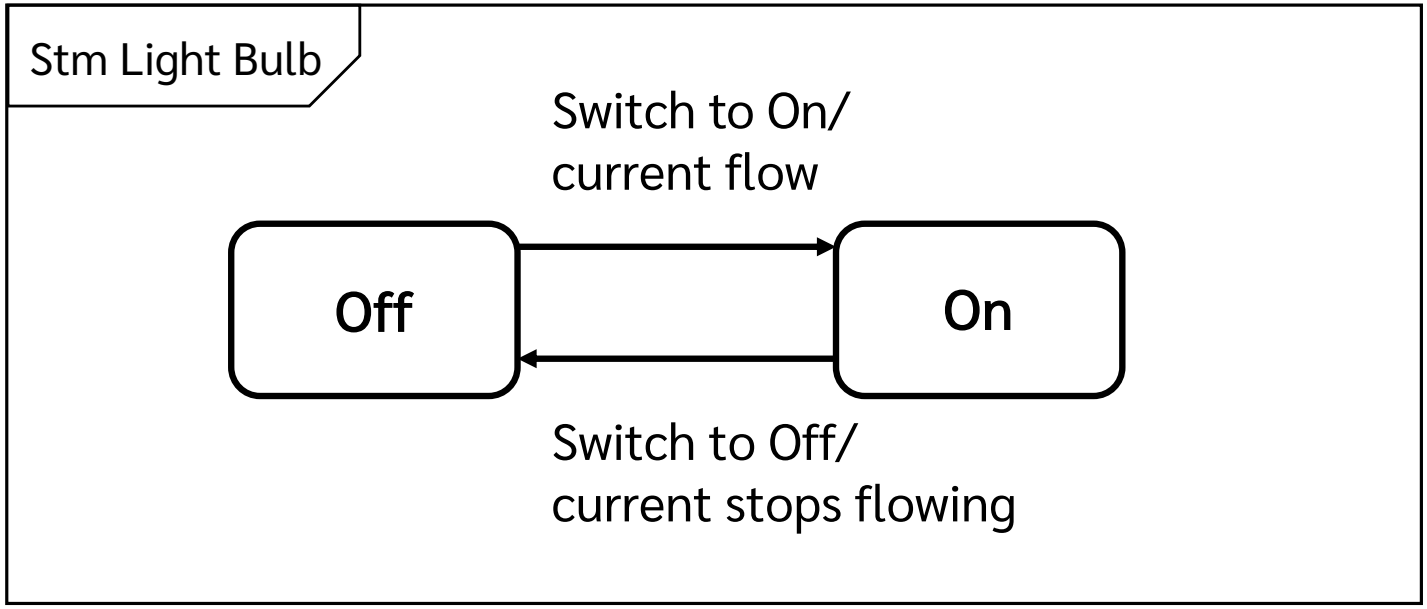
Nodes: States of the system (สี่เหลี่ยมมุมมน)

Edges: Transitions between states (from ⟶ to)

Edge Attribute: Events (เงื่อนไขการเข้าหรือออกจาก state) และ **Actions** (กิจกรรมที่เกิดขึ้นหรือทำระหว่างเปลี่ยน state)



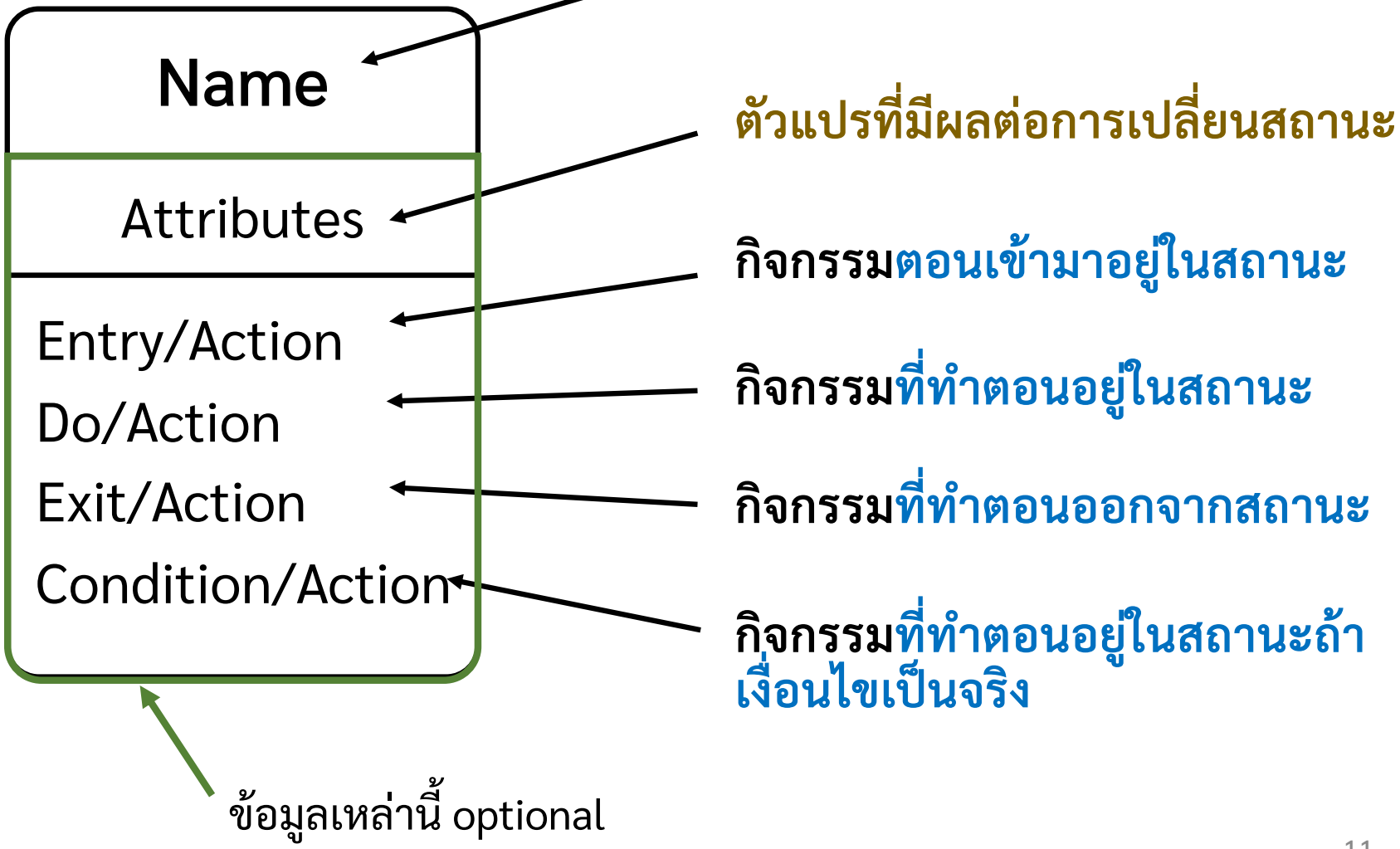
A Light Bulb



Event/ เหตุการณ์ (Switch to On)
Action/ กิจกรรมคือกระแสไฟฟ้าไหล (Current Flow)
State/ หลอดไฟเปลี่ยนสถานะเป็นเปิด

Event/เหตุการณ์ (Switch to Off)
Action/กิจกรรมคือกระแสไฟฟ้าหยุดไหล
State/หลอดไฟเปลี่ยนสถานะเป็นปิด

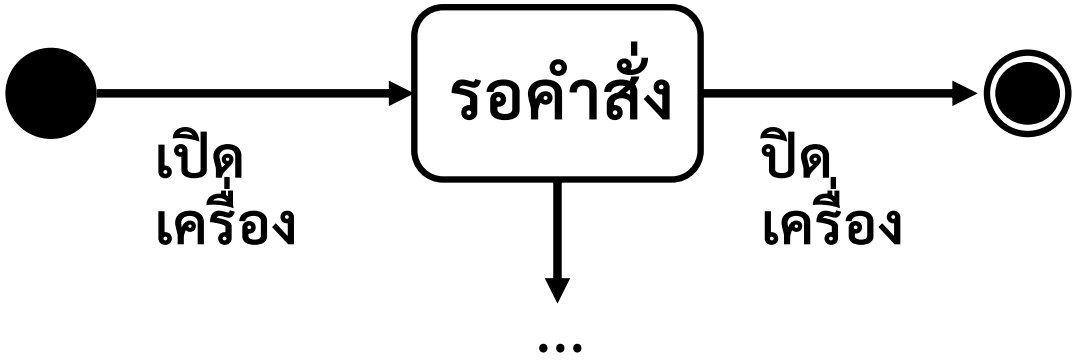
States


ชื่อของสถานะ
Name
ตัวแปรที่มีผลต่อการเปลี่ยนสถานะ
Attributes
กิจกรรมตอนเข้ามาอยู่ในสถานะ
Entry/Action
กิจกรรมที่ทำตอนอยู่ในสถานะ
Do/Action
กิจกรรมที่ทำตอนออกจากสถานะ
Exit/Action
กิจกรรมที่ทำตอนอยู่ในสถานะถ้าเงื่อนไขเป็นจริง
Condition/Action
ข้อมูลเหล่านี้ optional

Example: Clock

Example: State *Waiting for User Input*

Starting State and Ending State



Transition (การเปลี่ยนสถานะ)

- แสดงให้เห็นถึงเหตุการ์ณที่ทำให้ระบบเปลี่ยนจะสถานะหนึ่งไปอีกสถานะหนึ่ง
- แทนด้วยลูกศร ลากจากสถานะเริ่มต้น (from state) ไปยังสถานะใหม่ (to state)
- ข้อมูลที่เขียนอยู่บนลูกศรหมายถึงเหตุการ์ณ มีรูปแบบคือ

ทั้งนี้จะเขียนแค่ Event ก็ได้

Example: Transition ของการส่งการบ้าน Online

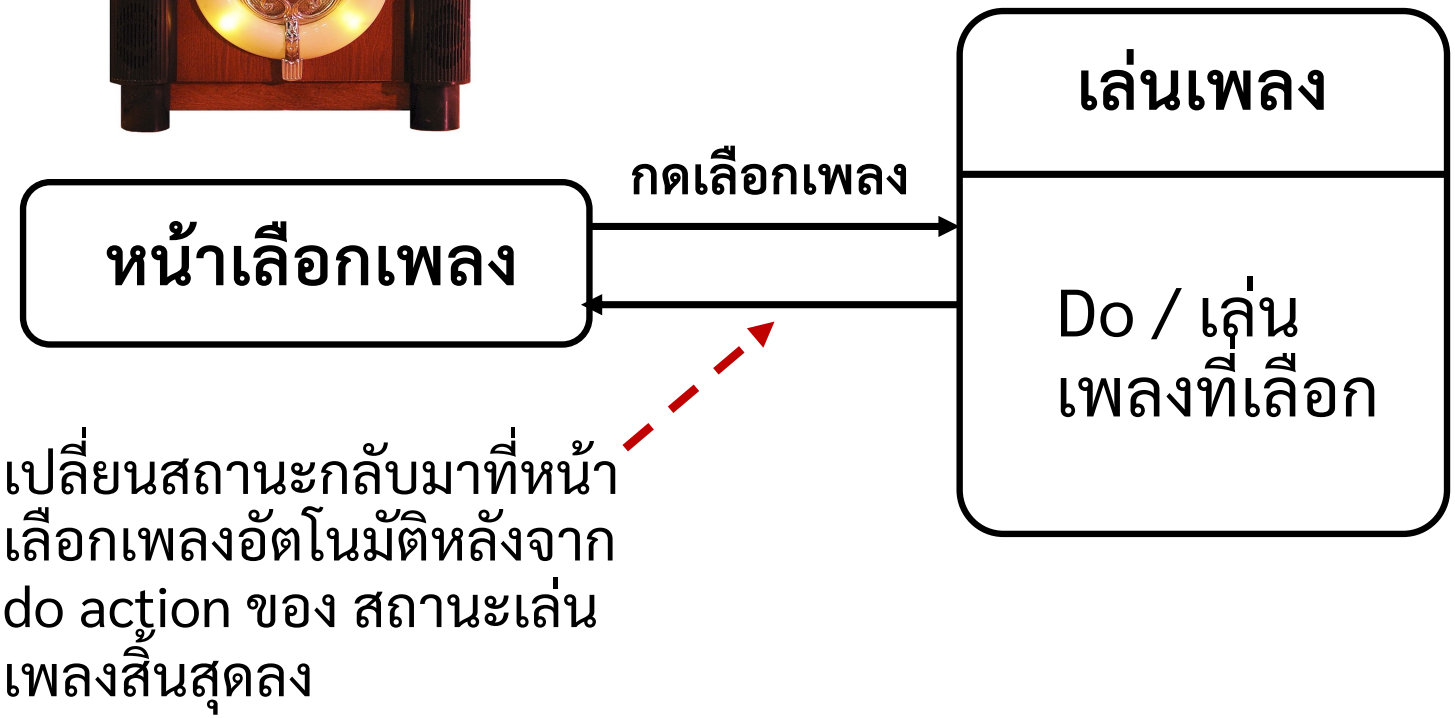
Steps ของการเกิดการเปลี่ยนแปลงสถานะ

1. An element is in the source state
2. An event occurs
3. An action is performed
4. The element enters a target state

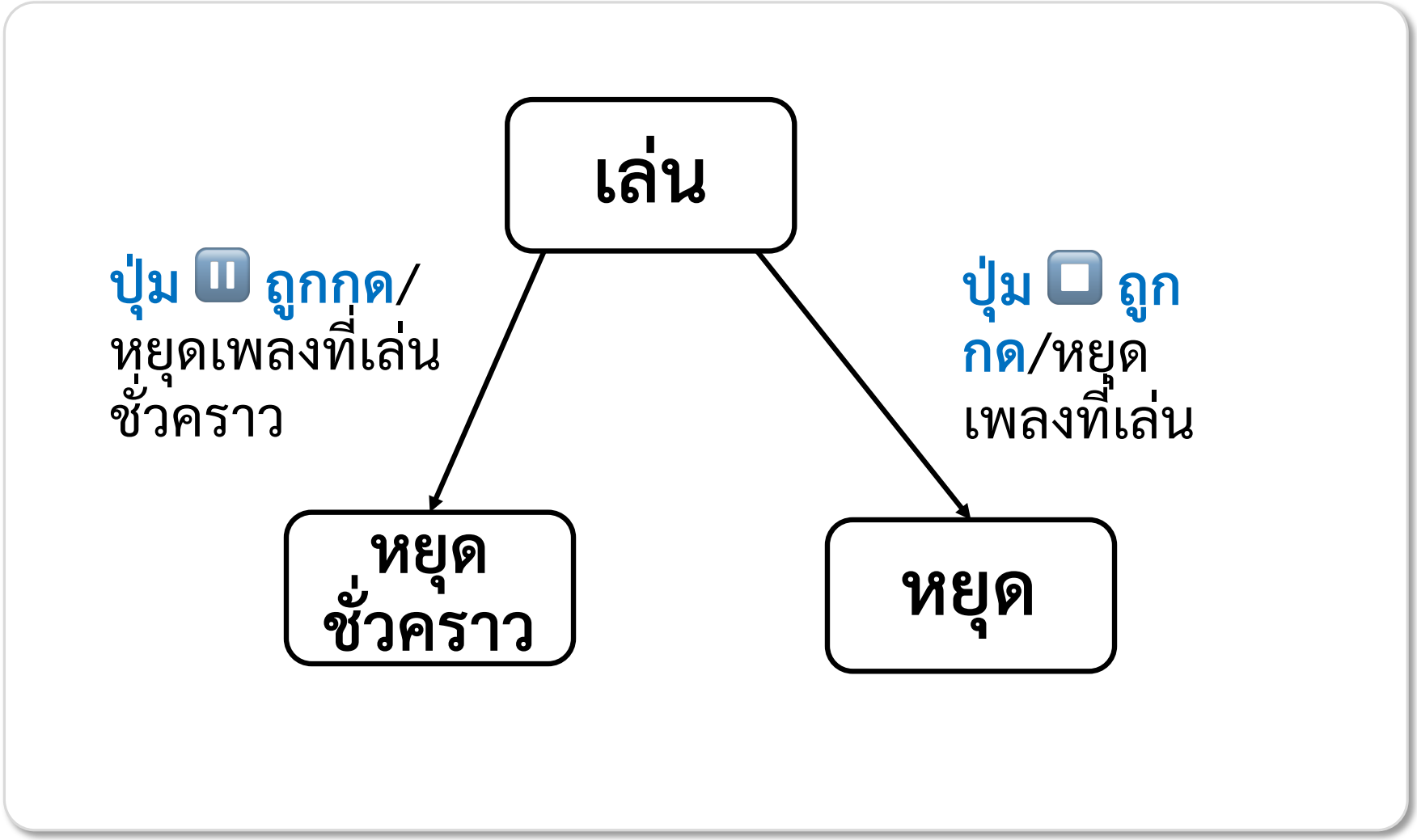
ถ้าการเปลี่ยนสถานะไม่มี event หรือ action กำกับเราจะเรียกมันว่า automatic transitions หรือ**การเปลี่ยนสถานะอัตโนมัติ**

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-state-machine-diagram/

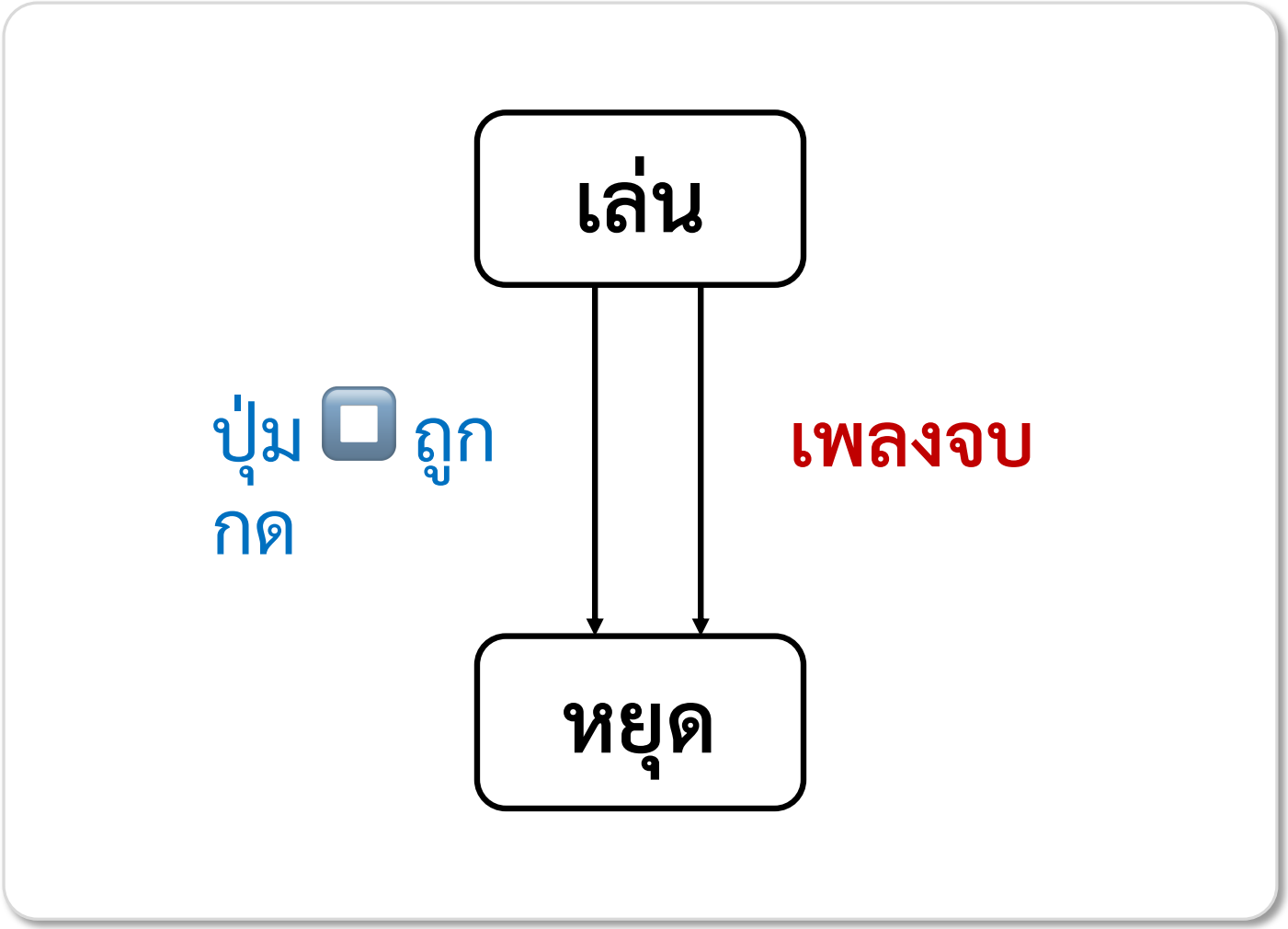
Example: Music Jukebox



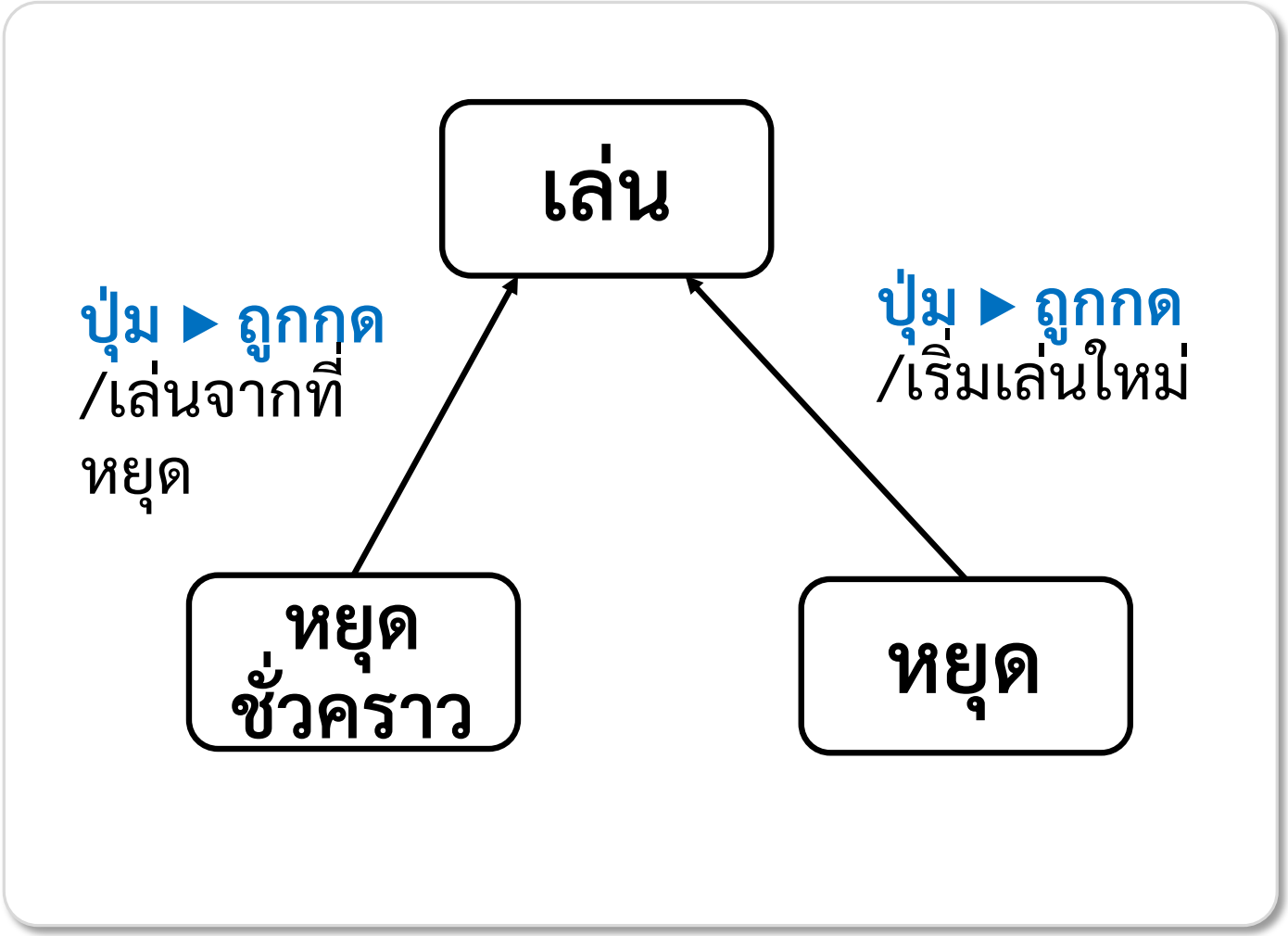
Different Events Transition to Different States



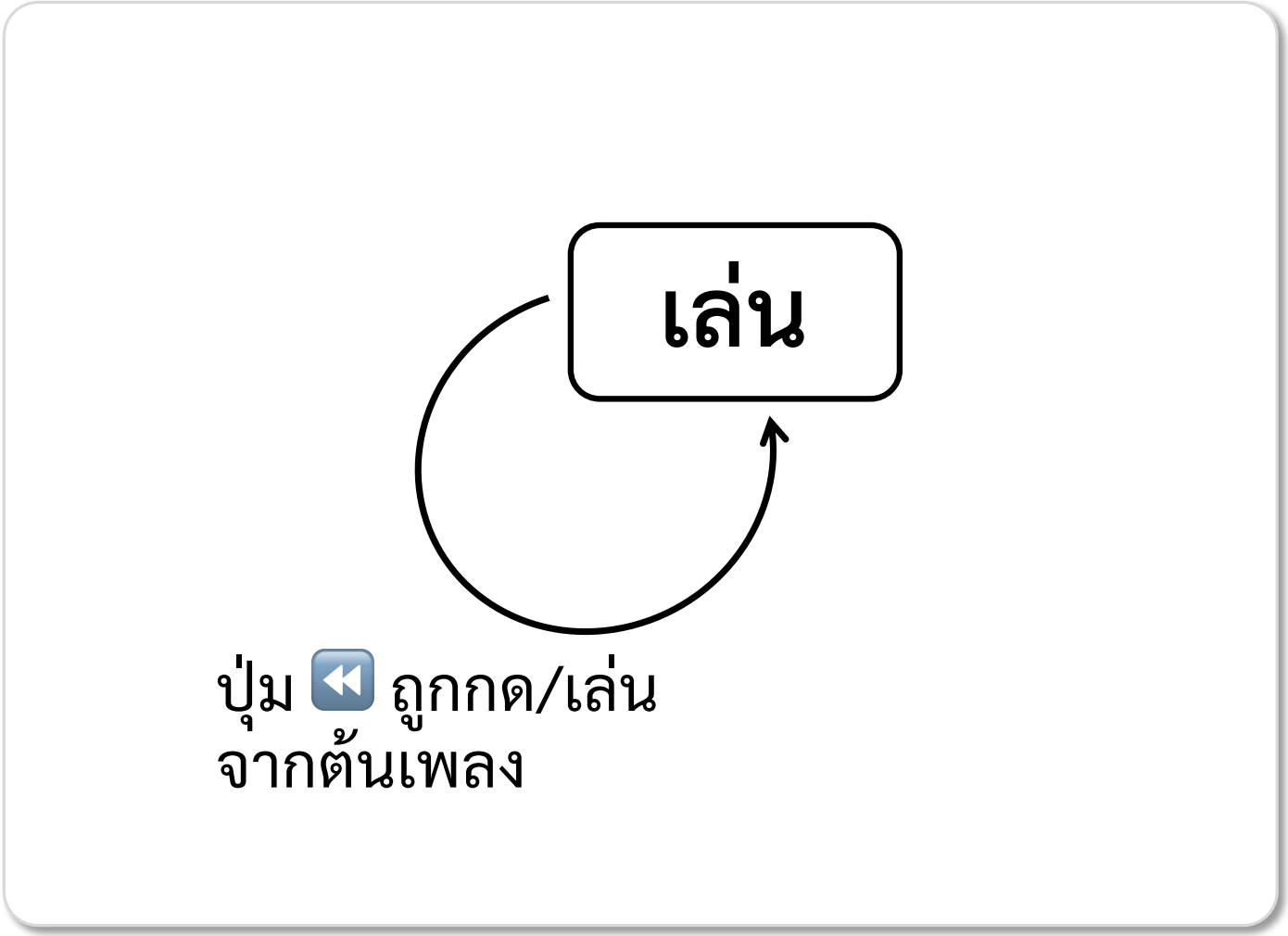
Different Events Cause Transitions to the Same State



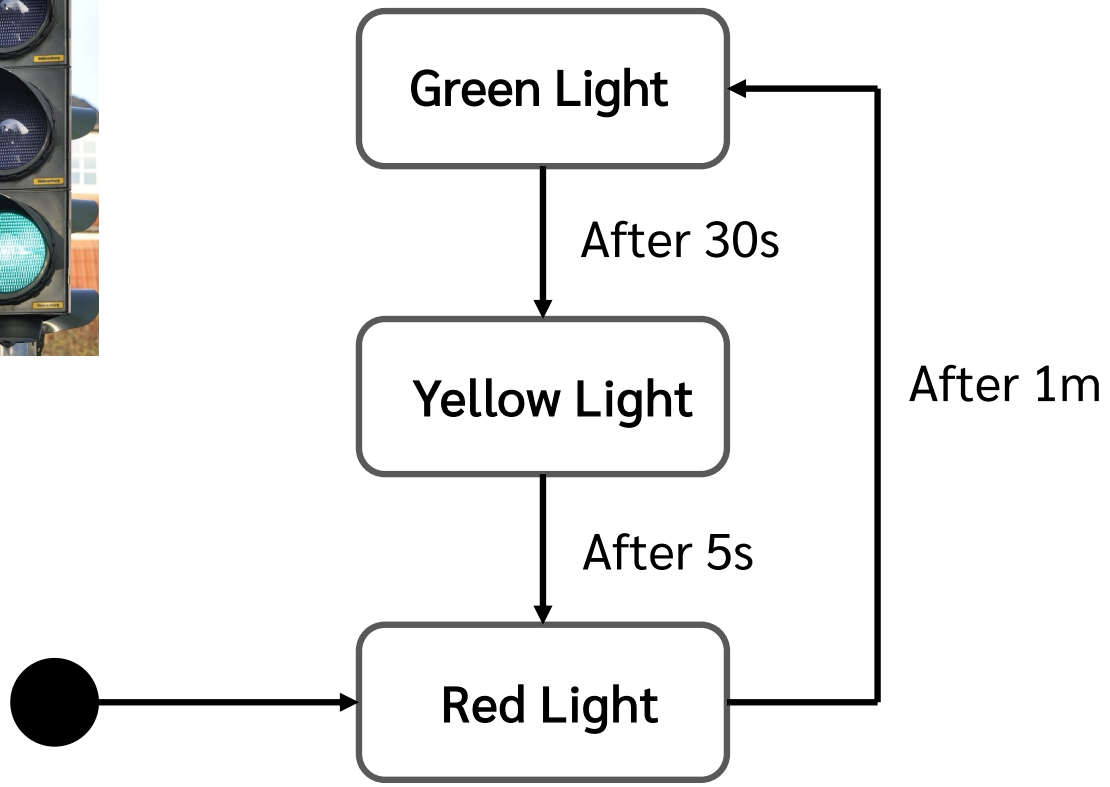
The Same Event Transitions From Different States to the Same State



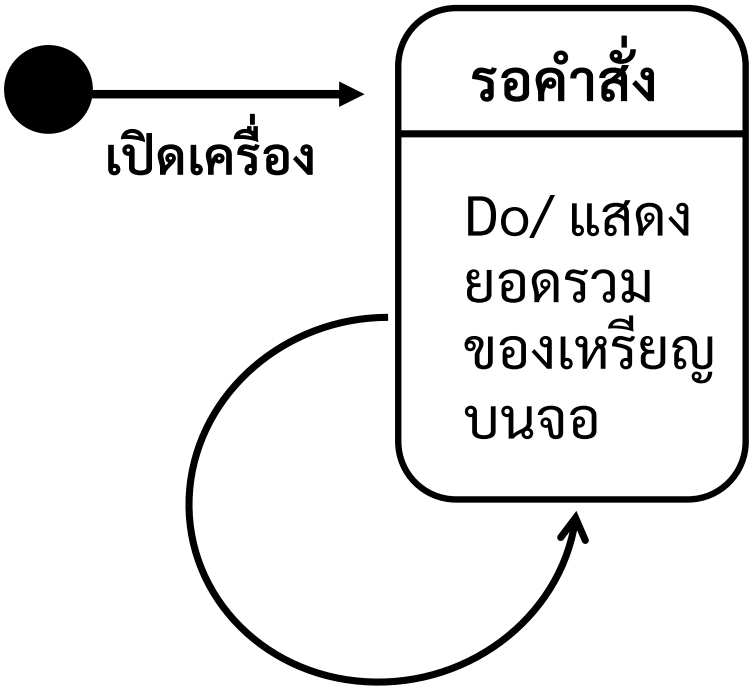
An Event Causes an Action, but No State Transitions



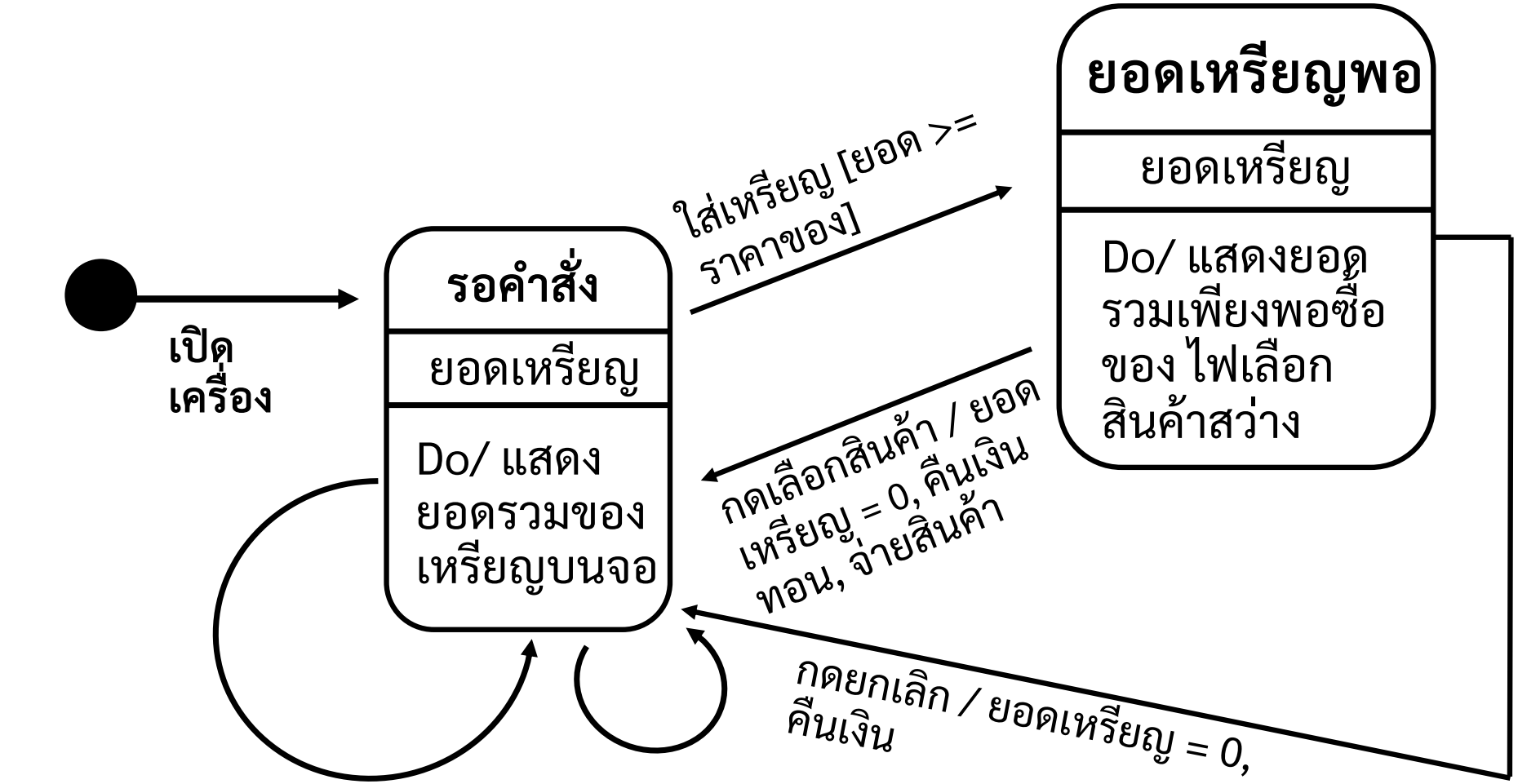
Example: Traffic Light with time-outs



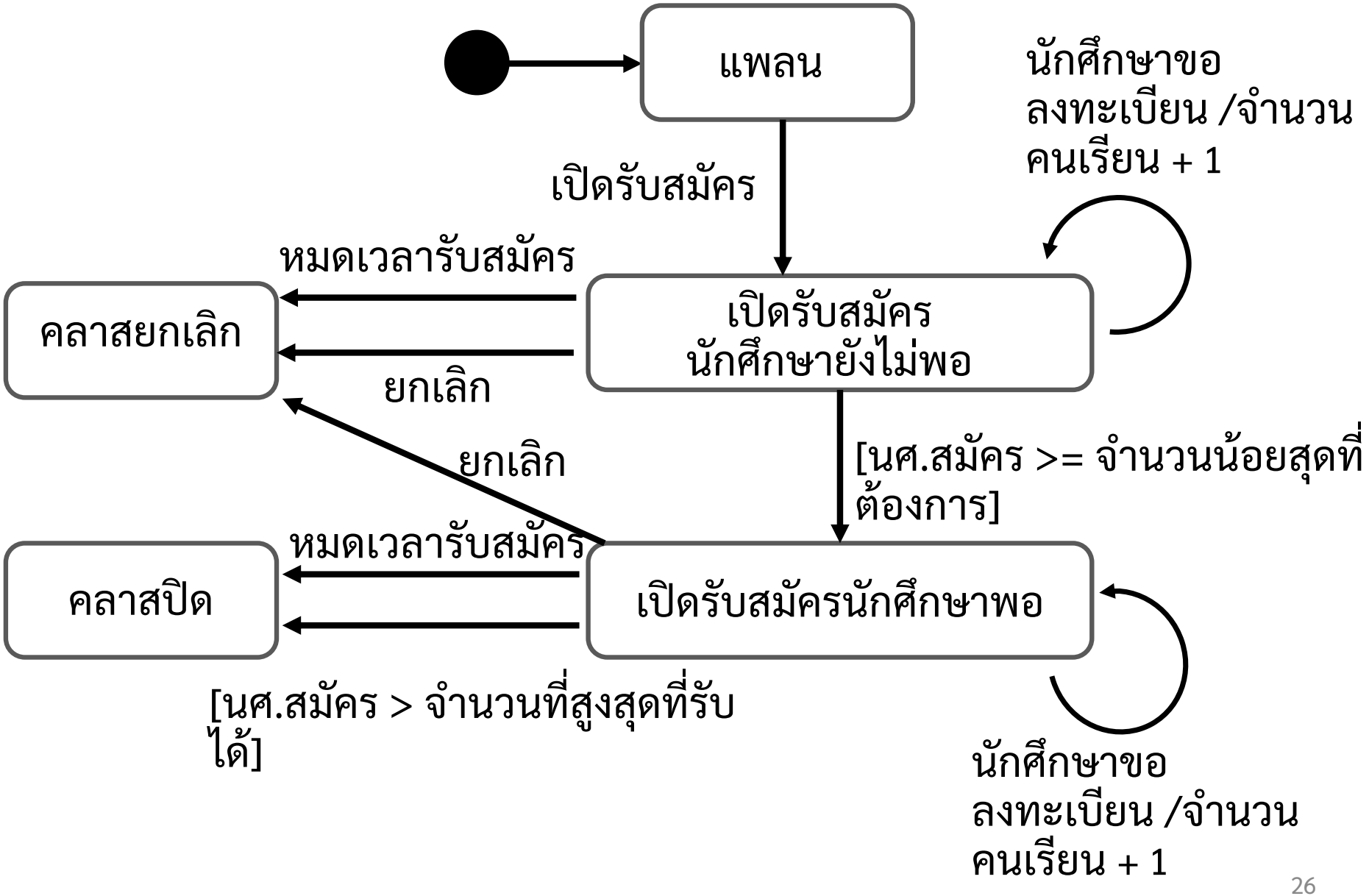
Example: A Vending Machine (เครื่องจำหน่ายสินค้าแบบหยอดเหรียญ)



Example: A Vending Machine (เครื่องจำหน่ายสินค้าแบบหยอดเหรียญ)



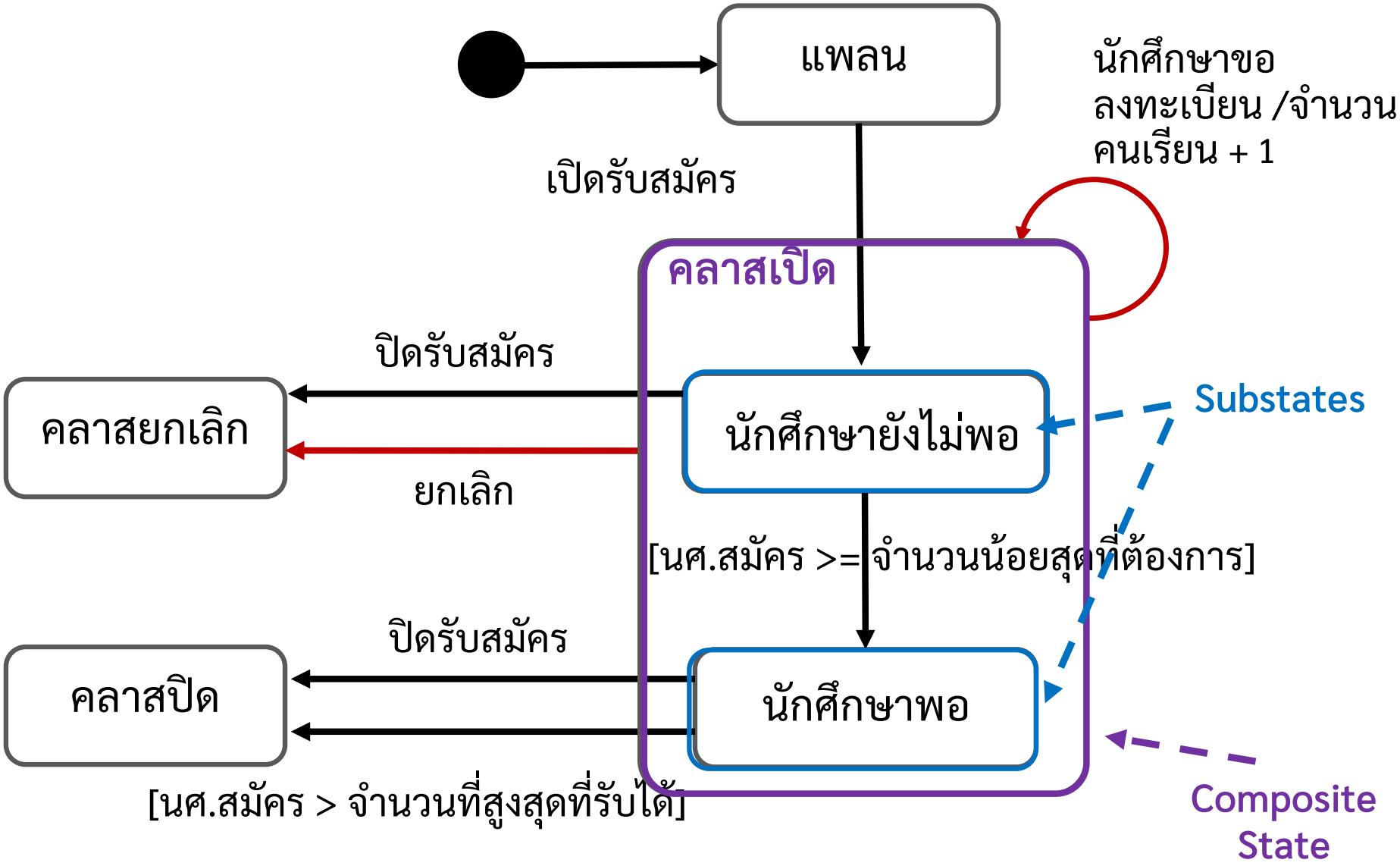
Example: A College Course



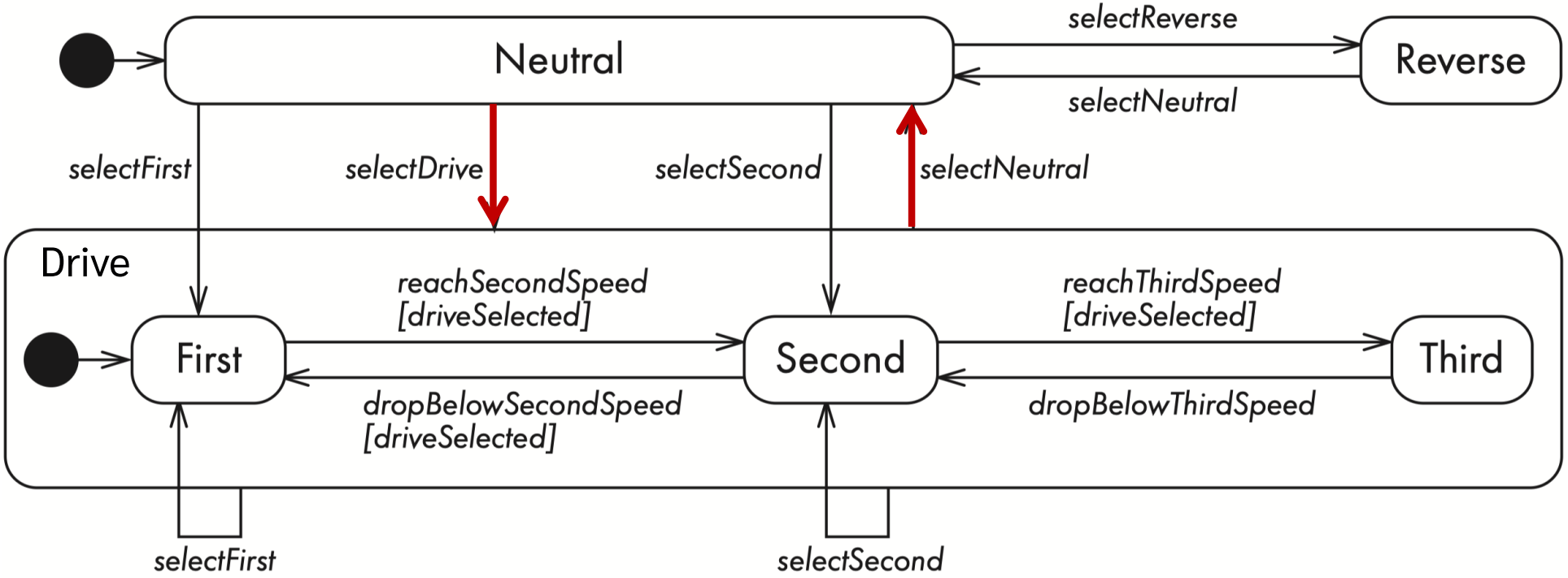
A Composite State and a Substate

- สถานะที่มีสถานะย่อยอยู่ภายใน เรียกว่า **Composite State**
- สถานะย่อยๆเรียกว่า **Substate**
- สถานะที่อยู่ภายในอาจจะมี**อย่างมากสุดหนึ่ง จุดเริ่มต้น และหนึ่งจุดสิ้นสุด**
  (แต่จะขึ้นอยู่กับว่า Composite State มีกี่ region (ขอบเขตของ substates))
- Composite State ที่มีมากกว่า 2 regions จะเรียกว่า Orthogonal Composite State
- เพื่อใช้ในรวมสถานะและตัวเชื่อมที่เกี่ยวข้องกันไว้ด้วยกันในหนึ่งสถานะ
- เป็นการลดการซับซ้อนของการแสดงสถานะต่างๆของระบบ

Example: A College Course (with Substate)



Example: Car Transmission (with Substate)

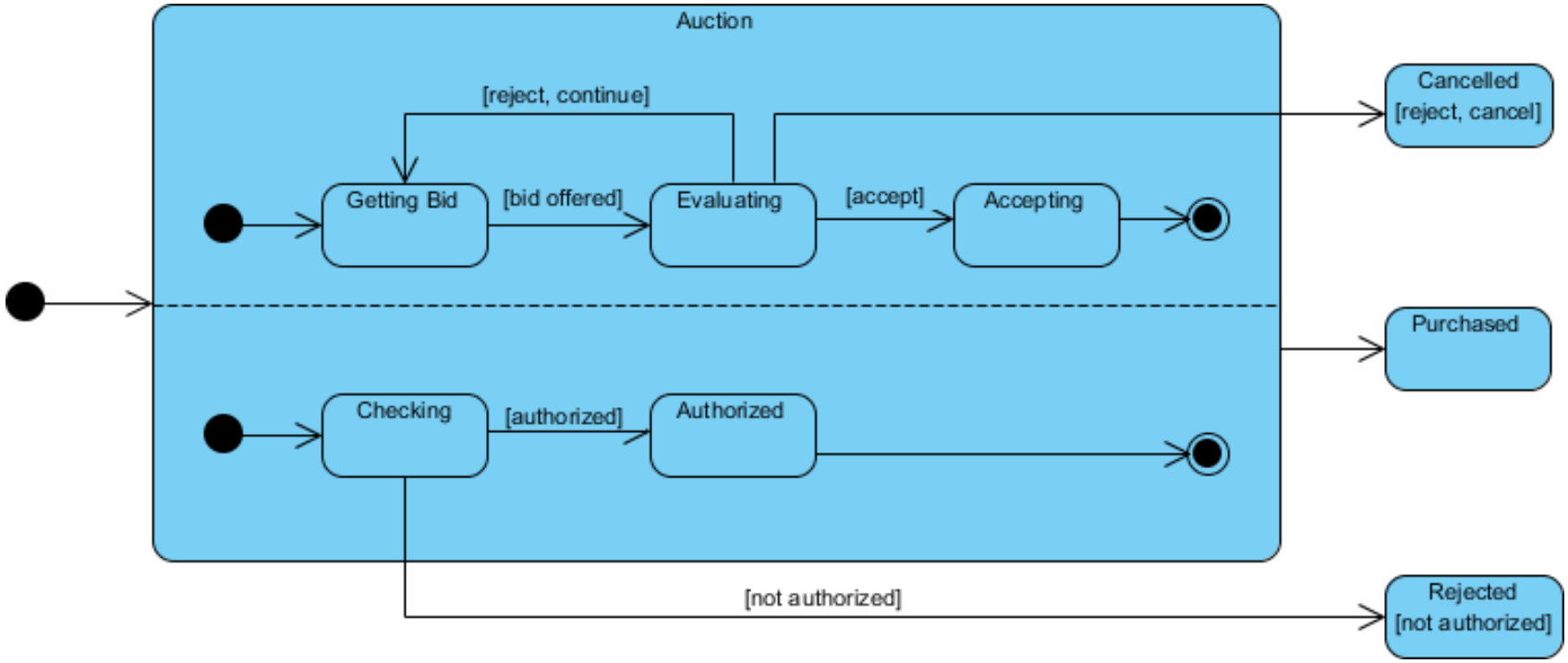


ขาดเกียร์ P (Park) ลองเก็บไป คิดดูว่าเราจะเอา diagram นี้มาแก้และเพิ่มมันอย่างไร

"Object-Oriented Software Engineering" by Lethbridge and Laganiere, 2nd Ed

Orthogonal Composite State: Concurrent

Composite State ของ “Auction” มีสอง regions ซึ่งสถานะต่างๆที่อยู่ในแต่ละ region จะแยกจากกันแต่สามารถเกิดขึ้นพร้อมกัน เราจะใช้เส้นประ (a dashed line) ในการแบ่ง region ของสถานะย่อยใน Composite State



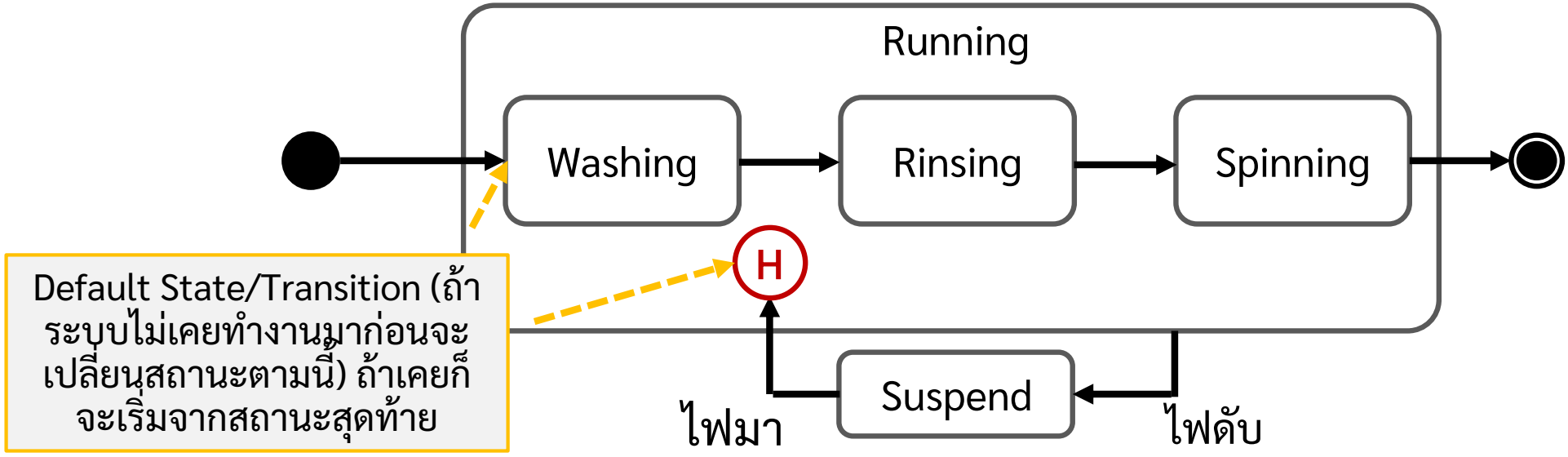
ตัวอย่างของระบบการประมูล ถ้าต้องการเปลี่ยนจาก Auction state เป็น Purchased state สถานะย่อยของทั้งสอง region ใน Auction ต้องไปถึง จุดสิ้นสุดของแต่ละ region ก่อน

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-state-machine-diagram/

History State

- โดยปรกติเมื่อออกจาก Composite State แล้วเข้ามันใหม่ ระบบจะเริ่มใหม่โดยอยู่ในสถานะย่อยเริ่มต้น Substate ของมัน *(เว้นแต่จะระบุไว้ว่าไม่ใช่)*

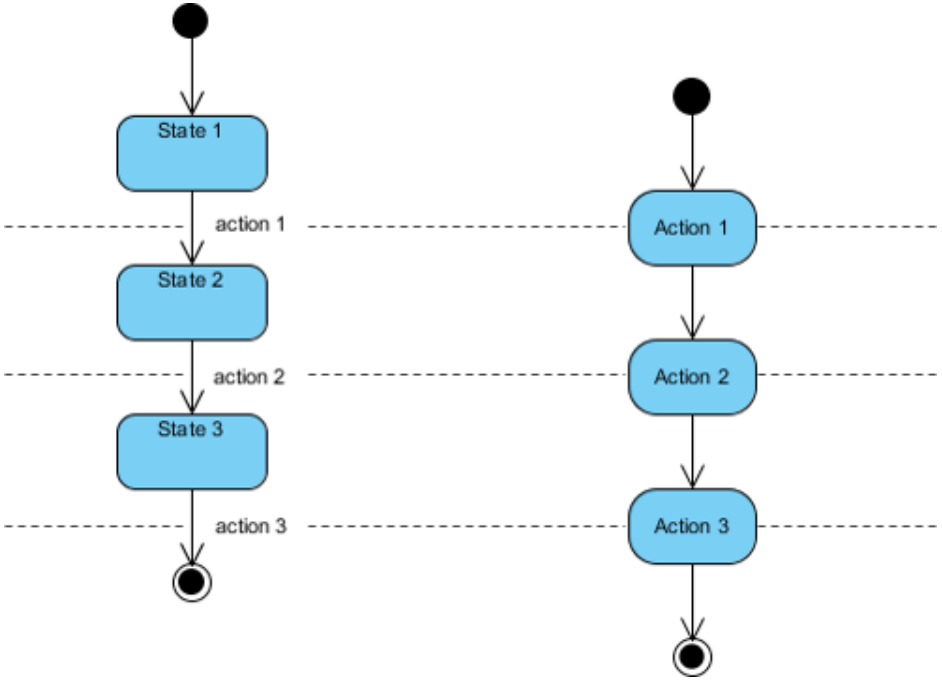
- History State ใช้บ่งบอกถึงการจำสถานะของระบบก่อนที่มันจะถูกรบกวนหรือมีการเปลี่ยนแปลงด้วยเหตุการ์ณบางอย่าง เมื่อมันกลับเข้ามาใน composite state มันก็จะอยู่ในสถานะย่อยสุดท้ายที่ก่อนที่ระบบจะออกจาก composite state ไป



https://sparxsystems.com/resources/tutorials/uml2/state-diagram.html

Activity vs State Machine Diagrams

- ADs สามารถทำออกมาหรือลดรูปให้เป็น SMDs ได้โดยที่ ADs มันมี *notations* บางอย่างที่ SMDs ไม่มี
- ใน ADs, nodes หมายถึง action และต้องทำ action หนึ่งๆให้เสร็จก่อนถึงจะไปถึง action ถัดไปได้
- ใน SMDs, nodes หมายถึงสถานะต่างๆที่เป็นไปได้**ของ object ในคลาส**และ และจะเปลี่ยนสถานะจากสถานะหนึ่งไปอีกสถานะหนึ่งจากเหตุการ์ณบางอย่าง



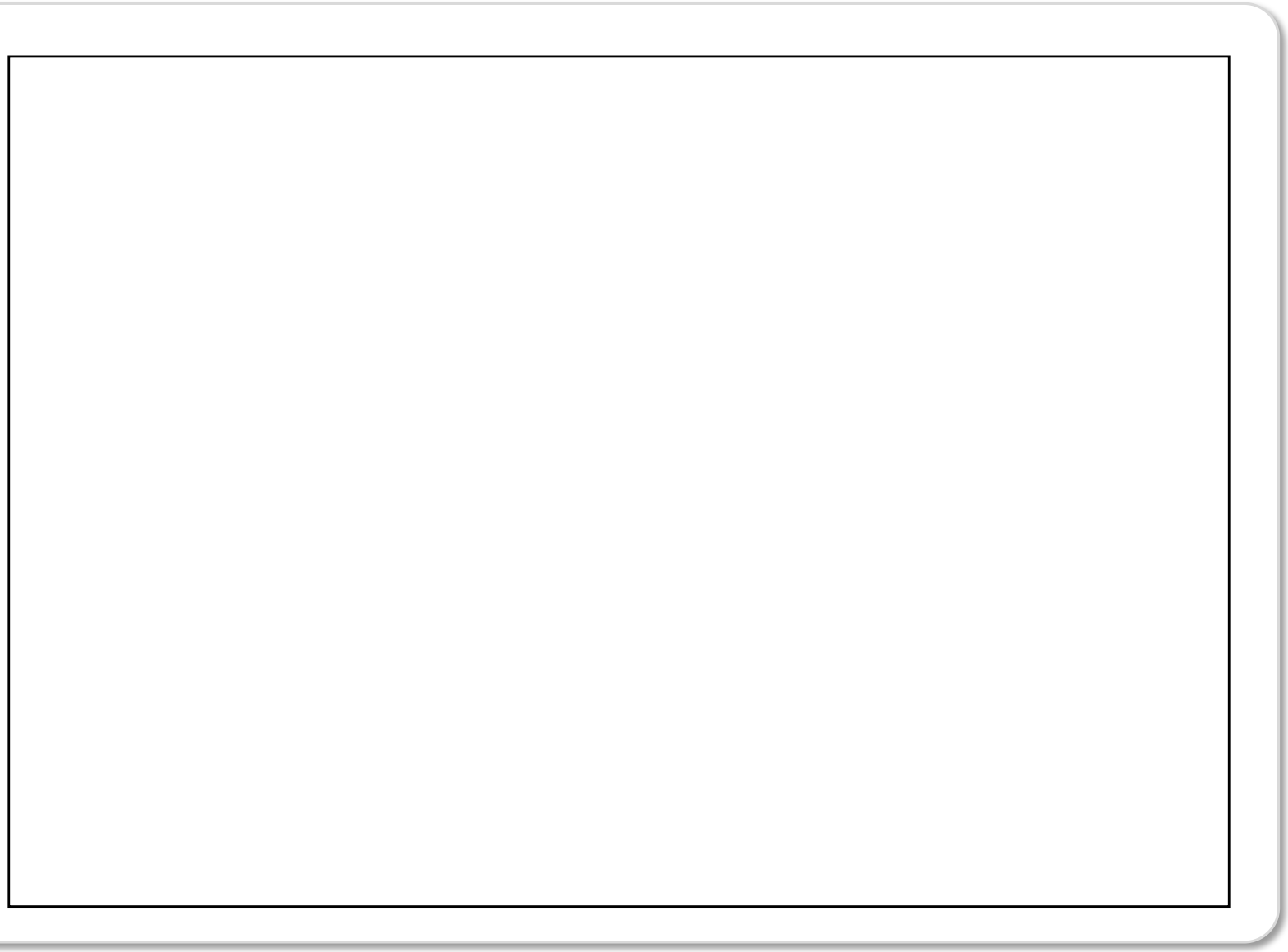
Action ทำให้เปลี่ยน State

Check Your Understanding

ให้นักศึกษารวมกลุ่มกับเพื่อนที่นั่งข้างๆ ลองเขียน State Machine Diagrams ของ ลิฟต์ (elevator/lift)

ให้นักศึกษารวมกลุ่มกับเพื่อนที่นั่งข้างๆ ลองเขียน State Machine Diagrams ของ เกม Tic-Tac-Toe (หรือ XO)

A Tic-Tac-Toe Game



Tic-Tac-Toe Game

Q: เราควรจะใส่ event กำกับ transition แต่ละเส้นของเกมส์ tic-tac-toe ด้านล่างนี้ว่าอะไรบ้าง

Summary

- State Machine Diagram and its notations
- Activity Diagrams vs State Machine Diagrams

Questions?