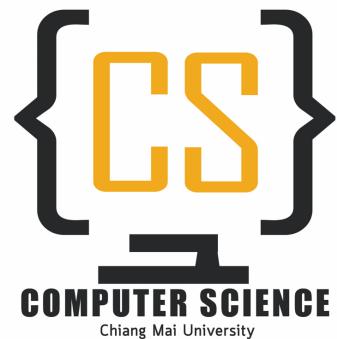


CS204362 – Object-Oriented Design

L4: Modeling Concepts

Kamonphop Srisopha



Faculty of Science, Chiang Mai University
คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่

Outline

- Models and Diagrams
 - Explain importance of them to software and system development
- Activity Diagrams

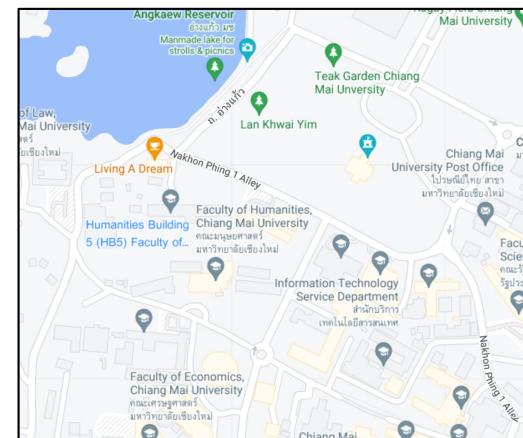
What is a Model (แบบจำลอง)?

- A model is an **abstract (simplified) representation** of something in the real world (or Imaginary), with a specific **purpose** and the **right level of detail**.
 - Like a map, a model represents something else.

แบบจำลองอธิบายลักษณะต่างๆของระบบอย่างสังเขป
โดยที่มันถูกสร้างเพื่อวัตถุประสงค์บางอย่าง และมี
รายละเอียดแค่ในระดับที่จำเป็นกับวัตถุประสงค์นั้นๆ



The Real world



A Model (for navigation purpose)

What are Models For? (เพื่ออะไร)

Communication

Visualization

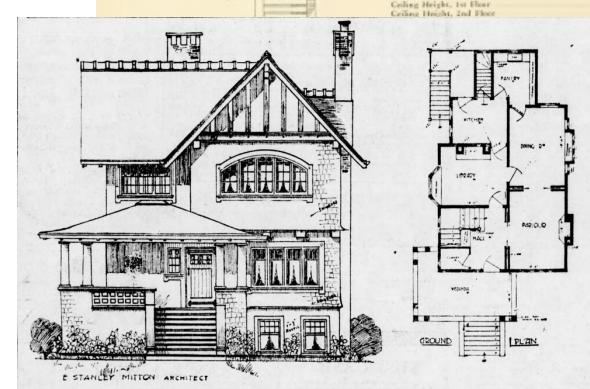
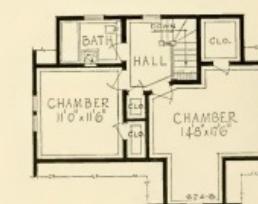
Verify

- เพื่อใช้ประกอบการสนทนาหารือ หรือ การคิดวิเคราะห์ ก่อนที่จะลงมือทำจริง
- เพื่อใช้ในระบบข้อกำหนดและความต้องการของระบบ เพื่อให้ผู้ที่มีส่วนได้ส่วนเสียกับระบบทั้งหมดเข้าใจ
- เพื่อใช้จัดการกับระบบที่ซับซ้อนให้เข้าใจง่าย
- เพื่อใช้สำรวจทางออกหรือวิธีแก้ต่างๆของปัญหาโดยไม่ต้องลงมือทำจริง
- เพื่อทำให้เราเห็นรายละเอียดของสิ่งต่างๆได้ชัดเจน



Six Rooms 28100 Cubic Feet
Design 624-B

A SHINGLED home with an unusually attractive exterior and a well-planned interior remarkably convenient for easy housekeeping. Notice the chamber and toilet on first floor.



Example of Models

Requirements Model

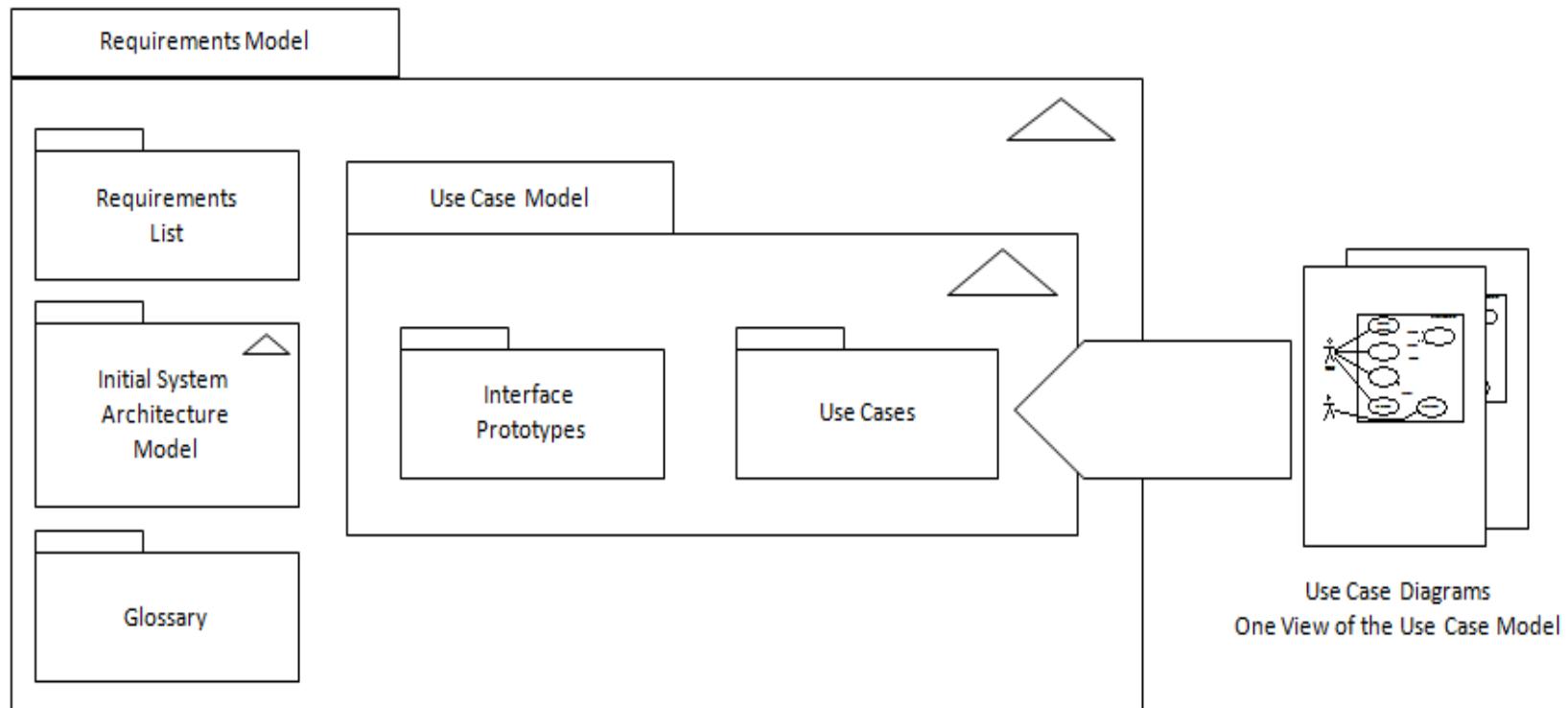
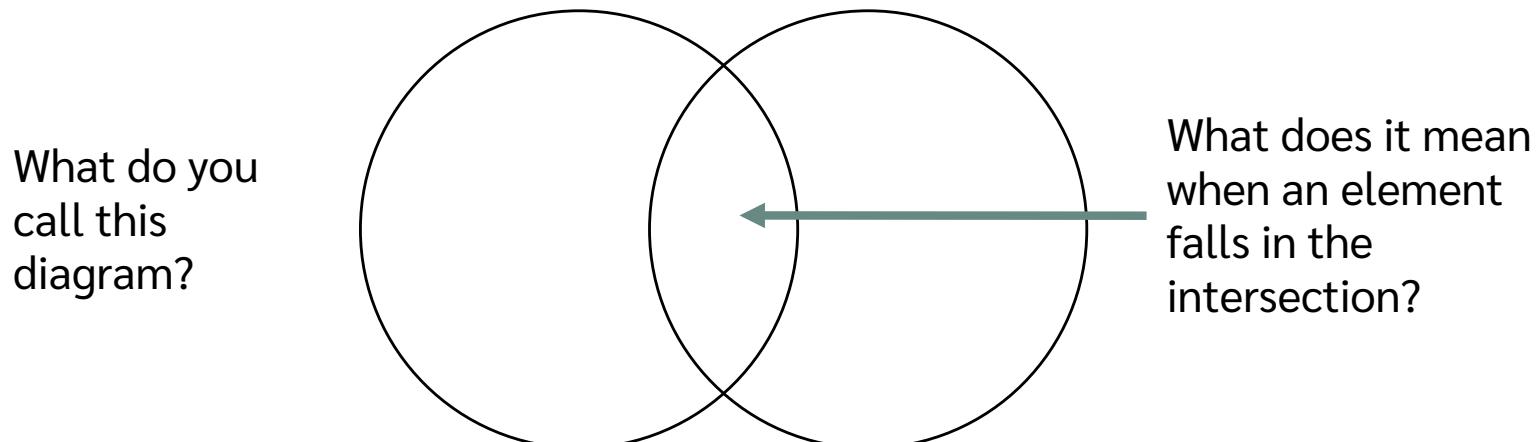


Figure 5.4 Illustration of a UML model and its relationship with one type of diagram (Use Case Diagram)

What is a Diagram?

- Diagrams are abstract shapes that are used to **visually or graphically** present **some part of a model** or of something from the real world.
- Diagrams follow **rules** or **standards**.
- The rules or standards make sure that different people will **interpret** the diagram in **the same way**.



Why should Diagrams Be Used?

การสื่อสารโดยใช้ภาษาพูดส่วนมากมักจะกำกับ

พ่อไปกรุงเทพฯ กับน้ำสองคน

เขารំรงทำงานใหលូវ

នំនែងແអបកិនខ្លួនឱ្យឈើ

Diagrams ទីផ្សារតាម rules และ standards នឹងធ្វើជាផ្លូវការដូចម៉ោងខាងក្រោម ដើម្បីពិនិត្យការសេវាការណ៍ទាំងអស់។

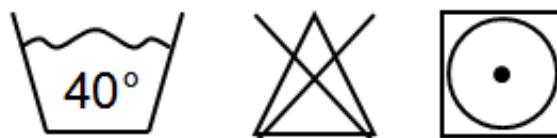


Figure 5.3 Example from a diagram standard.

Warm Water at
40 Celsius

Do not bleach

Tumble dry with
low heat

Example: A diagram of the tasks in producing a book

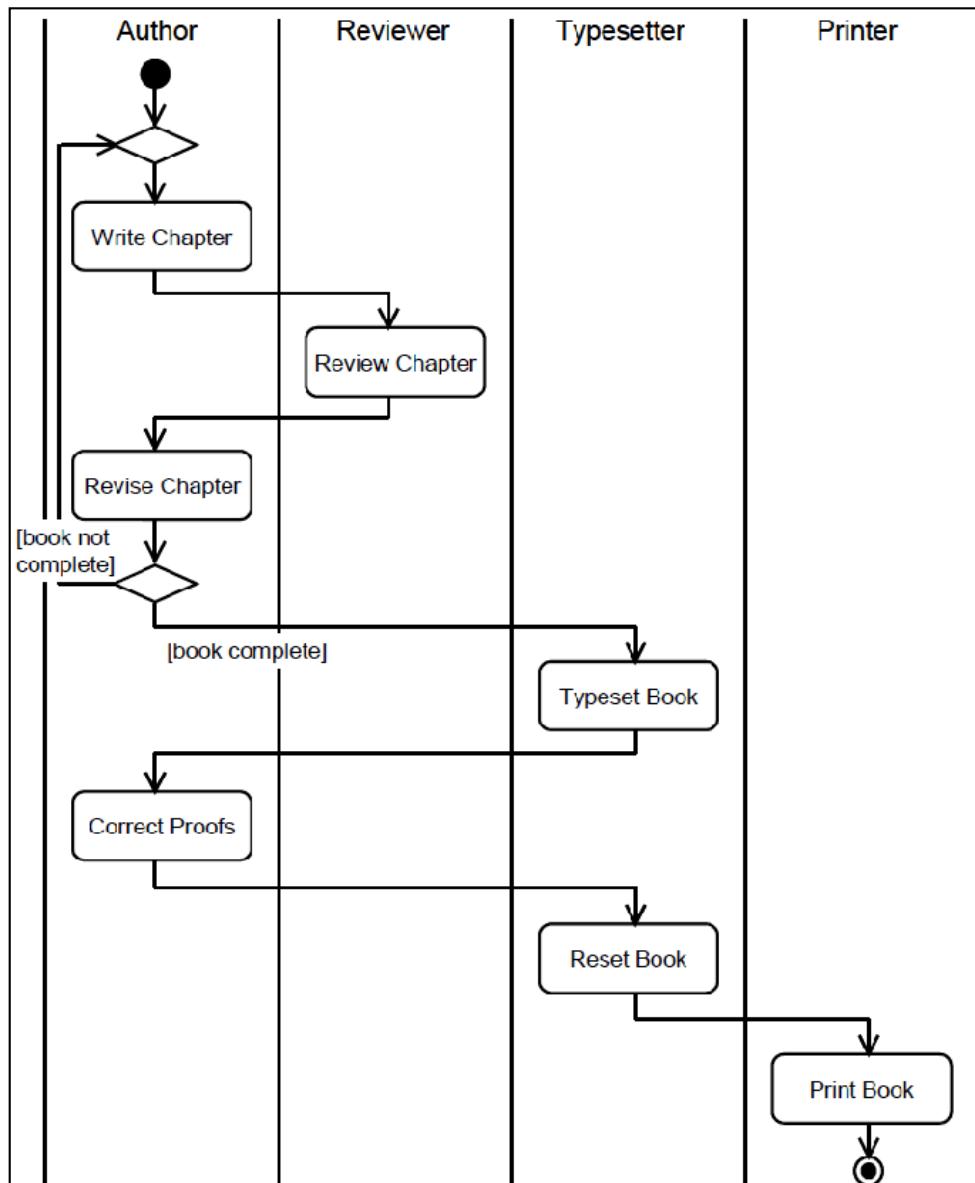
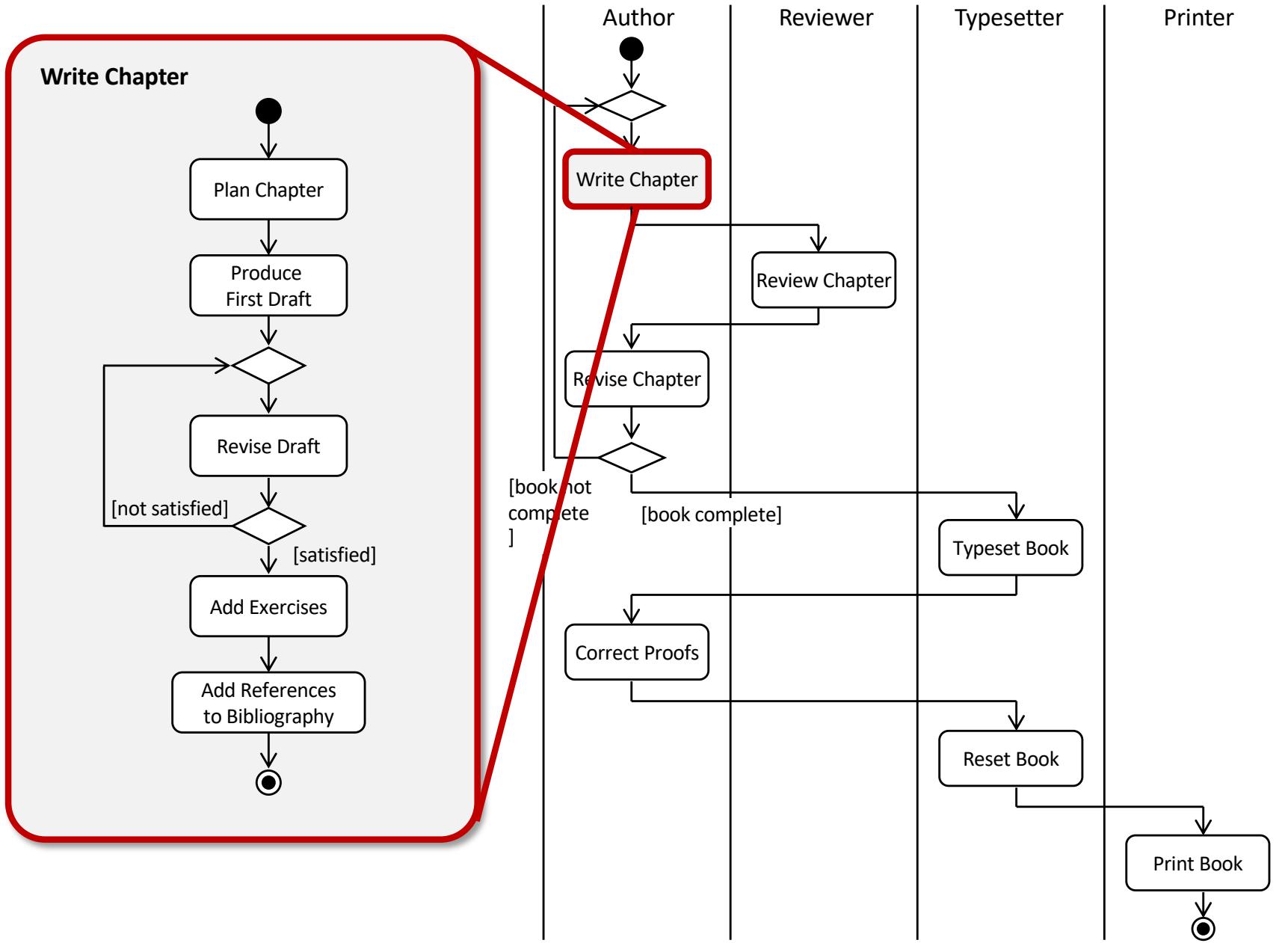


Diagram นี้แสดงขั้นตอน
ต่างๆในการผลิตหนังสือ
ทาง UML เรียก diagram นี้
ว่า **Activity Diagram**

The Right Level of Detail



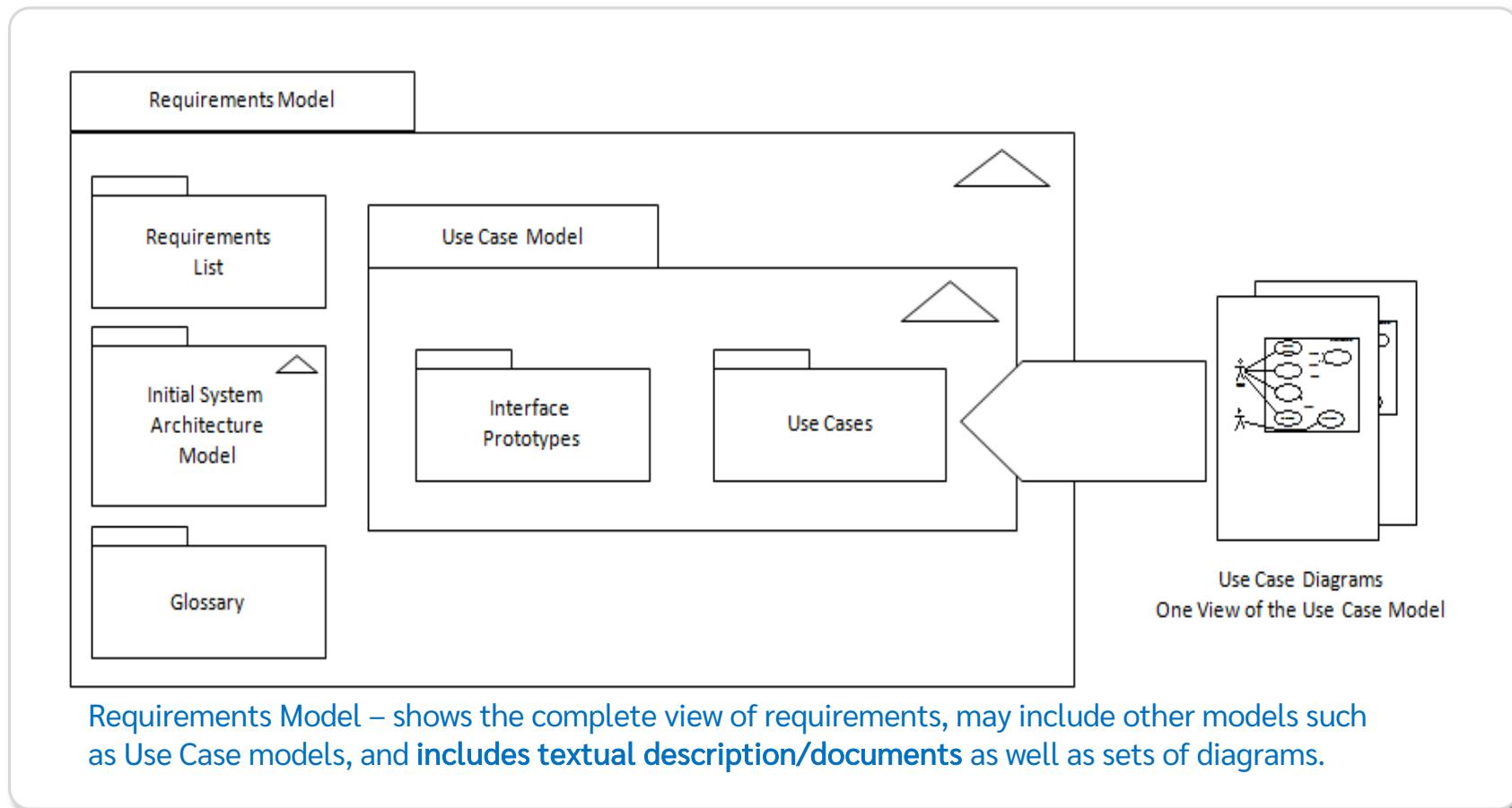
Models vs Diagrams

- Diagrams ใช้แสดงบางส่วนของระบบ (**visually or graphically**)
- Models แสดงมุ่งมองที่ครอบคลุมของระบบ ณ เวลาใดเวลาหนึ่ง และมุ่งมองโดยมุ่งมองหนึ่ง และไม่จำเป็นต้องเป็น graphics

Model ประกอบไปหนึ่งหรือหลายๆ diagrams ที่เกี่ยวข้องกันเพื่อให้ครอบคลุม แบ่งมุ่งต่างๆ และ model ยังรวมถึงข้อมูล และเอกสารสนับสนุนต่างๆด้วย

Relationships Between Models and Diagrams

Use Case Diagrams เป็นหนึ่งมุ่งมองใน Use Case Model ซึ่งอยู่ภายใต้ Requirements Model อีกที

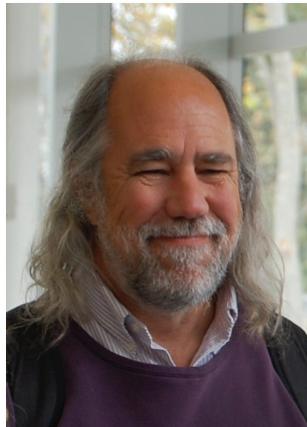


Unified Modeling Language (UML)

What is UML?

Unified Modeling Language (UML)

เป็นภาษารูปแบบมาตรฐานที่ใช้เพื่อถ่ายทอดความเข้าใจของระบบและ
จำลองระบบในด้านต่างๆให้ออกมาเป็นแผนภาพ เชื่อว่ามีความละเอียด
จนสามารถจำลองระบบทุกรอบแบบให้ออกมาเป็นแผนภาพได้ ภาษาที่ถูก^{พัฒนาขึ้น}เพื่อใช้สำหรับการวิเคราะห์และออกแบบระบบเชิงวัตถุ คิดค้น^{โดย} Grady Booch, Ivan Jacobsen และ Jim Rumbaugh



Grady Booch

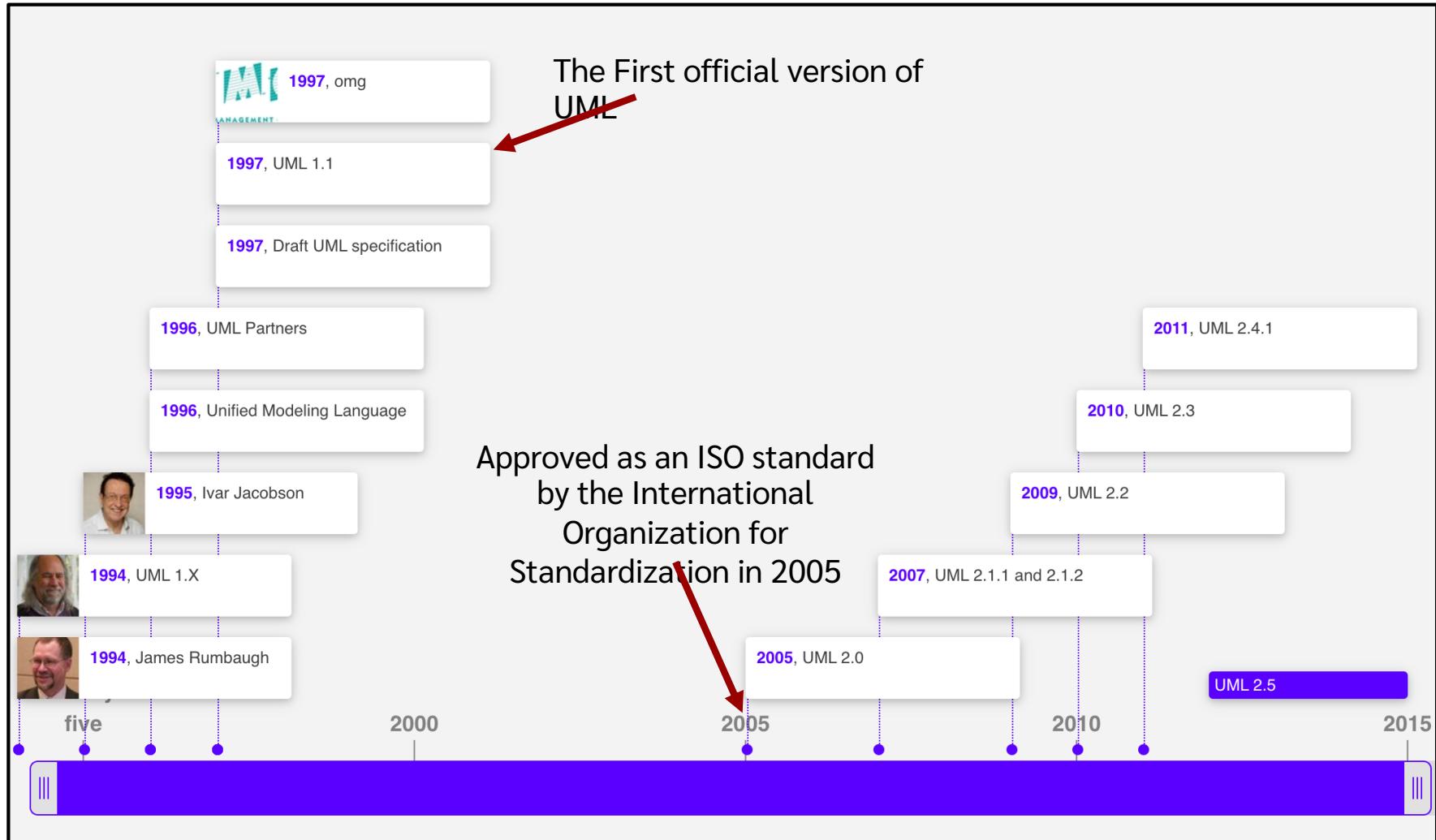


Ivan Jacobsen



Jim Rumbaugh

History of UML



The latest UML version is 2.5.1 which was released in December 2017.

Models in UML

- A system is the overall thing that is being modelled
- A subsystem is a part of a system consisting of related elements
- A model is an abstraction of a system or subsystem from a particular perspective
- A model is complete and consistent at the chosen level of abstraction

Models in UML

- Different models present different views of the system, for example:
 - use case view
 - design view
 - process view
 - implementation view
 - deployment view

(Booch et al., 1999)

Developing Models

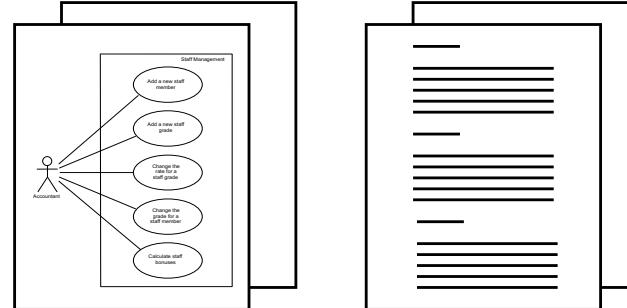
- โมเดลที่เราสร้างขึ้นตอนพัฒนาระบบและจะถูกเปลี่ยน (update) ไปตามความก้าวหน้าของการพัฒนาระบบในหลายๆ ด้าน (**iterative approach**) เช่น
 - **Abstraction**—they become more concrete (เห็นเป็นรูปธรรมมากขึ้น)
 - **Formality**—methods, attributes, and constraints will become more formally defined (ดูเป็นระเบียบแบบแผนเป็นพิธีมากขึ้น)
 - **Level of detail**—additional detail is added as understanding improves (รายละเอียดจะเพิ่มขึ้นเมื่อเข้าใจระบบมากขึ้น)

Development of the Use Case Model Through Successive Iterations

Iteration 1

Obvious use cases.

Simple use case descriptions.

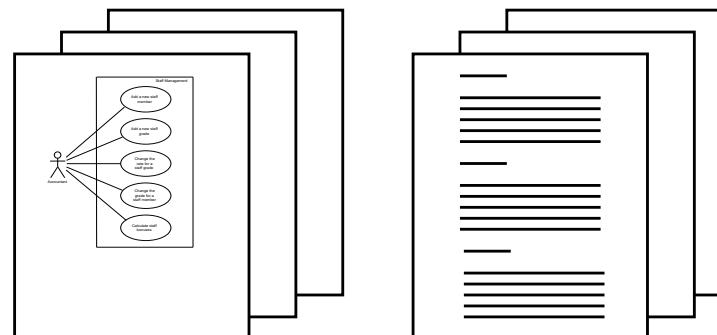


Iteration 2

Additional use cases.

Simple use case descriptions.

Prototypes.

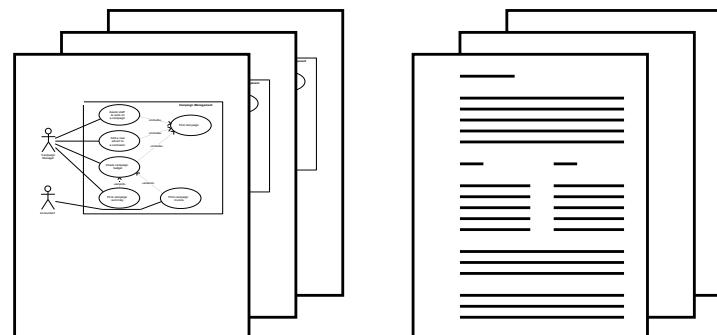


Iteration 3

Structured use cases.

Structured use case descriptions.

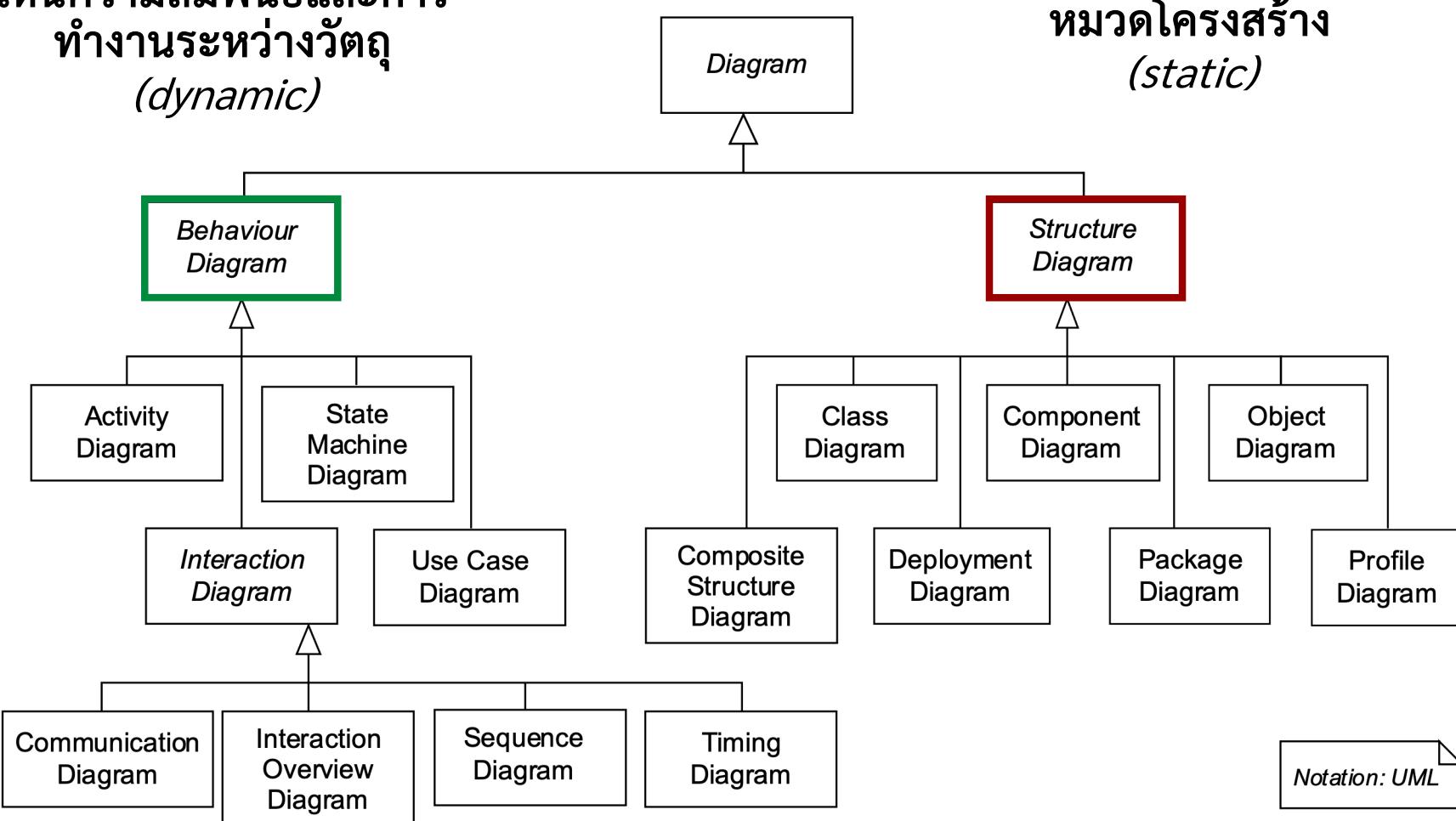
Prototypes.



Iteration n

Modeling a System using UML Diagrams

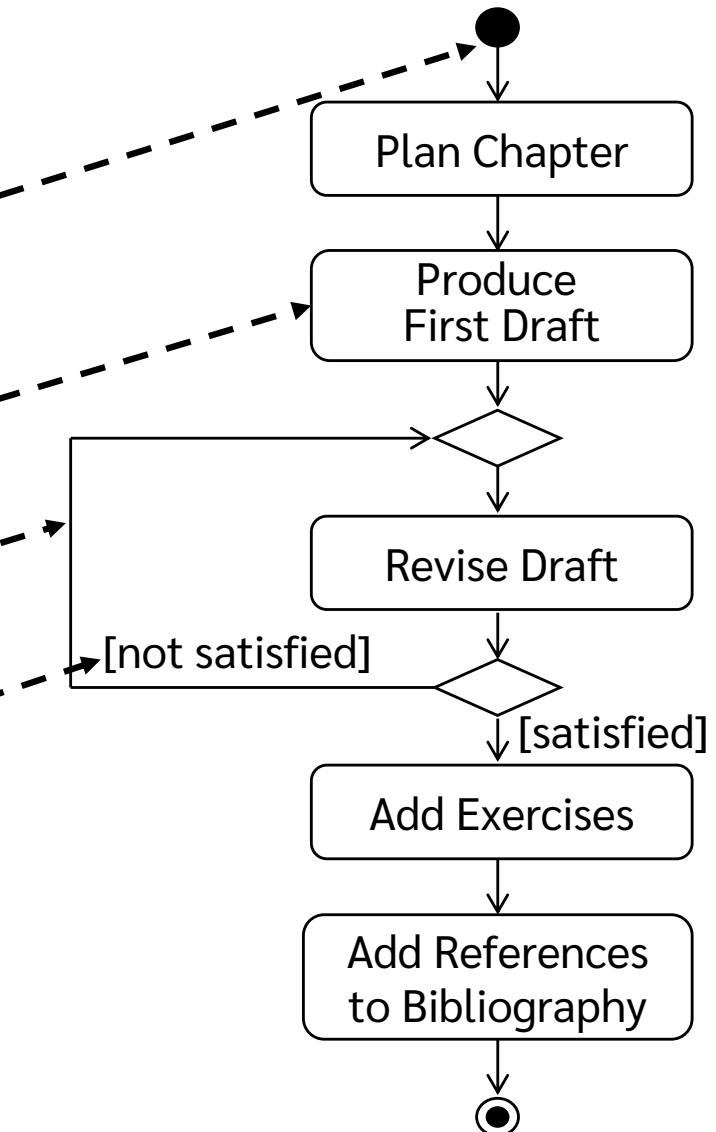
หมวดพฤติกรรมแสดงให้
เห็นความสัมพันธ์และการ
ทำงานระหว่างวัตถุ
(dynamic)



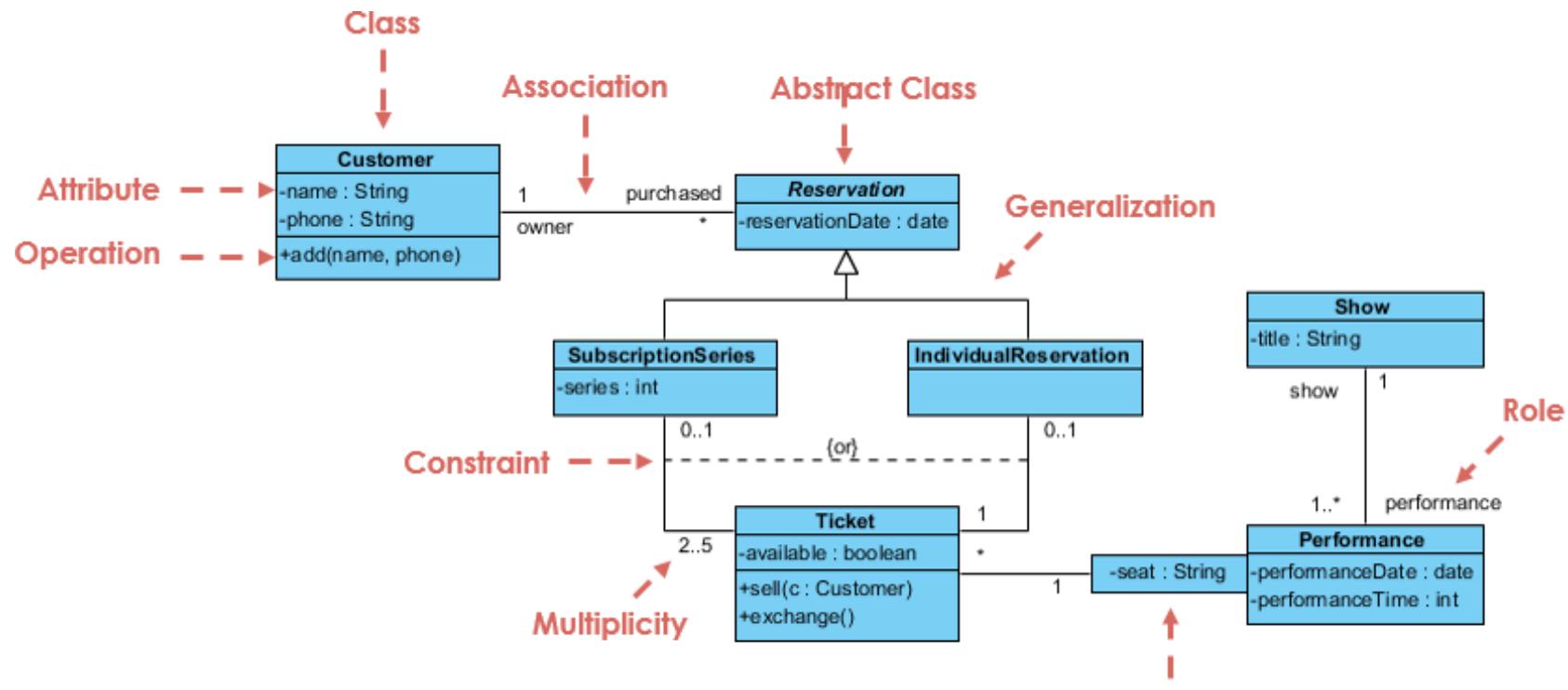
Diagrams in UML

UML diagrams consist of:

- Icons
- Symbols
- Paths
- Strings

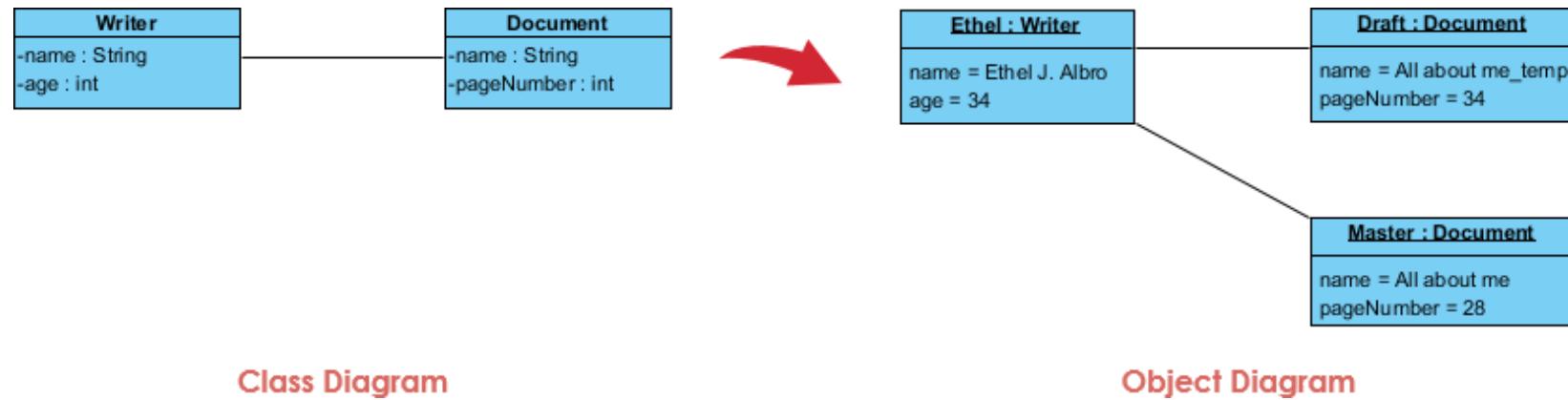


Example of UML diagrams (*Structure*)



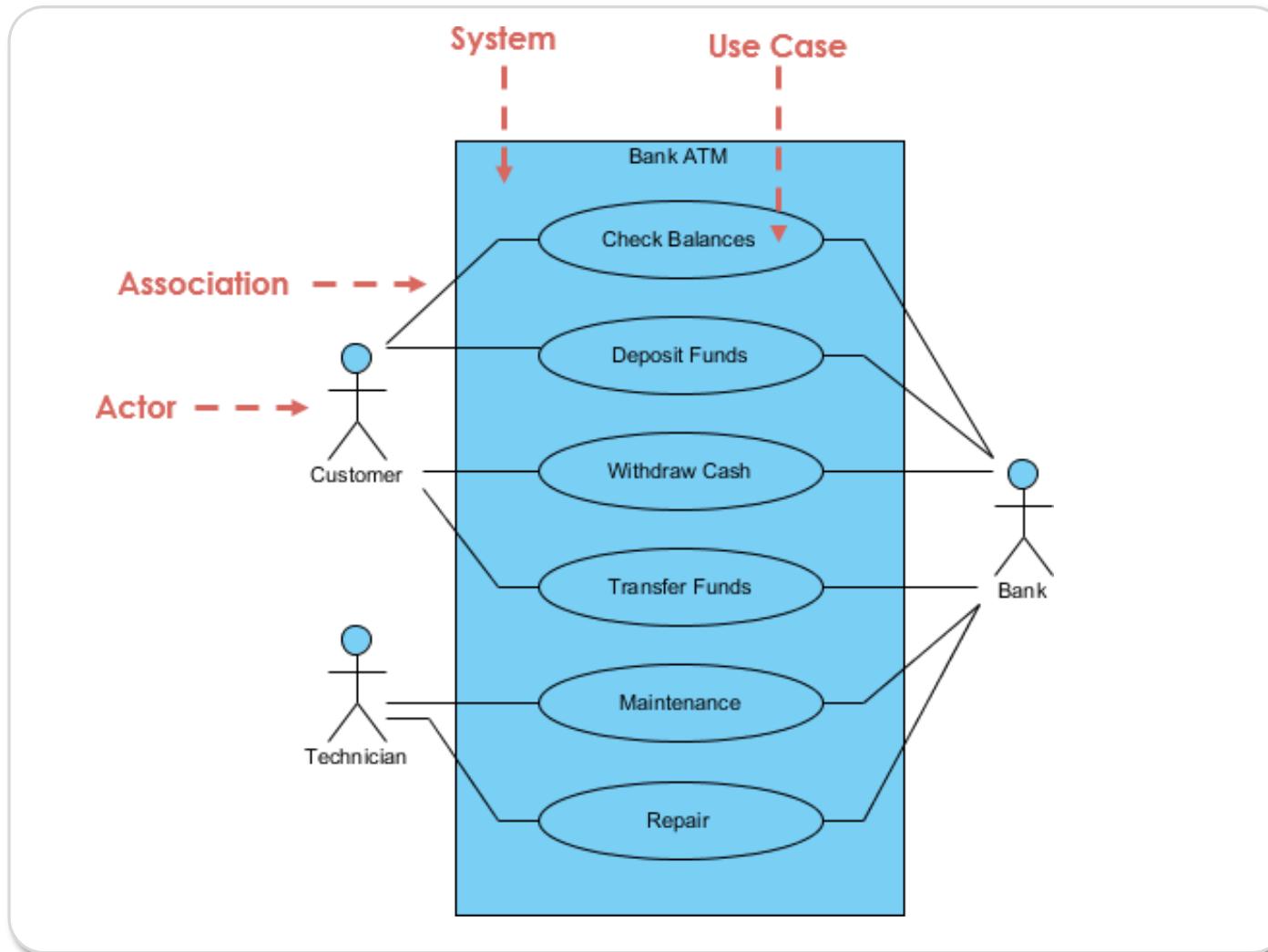
Class Diagram
(Ticket Selling System)

Example of UML diagrams (*Structure*)



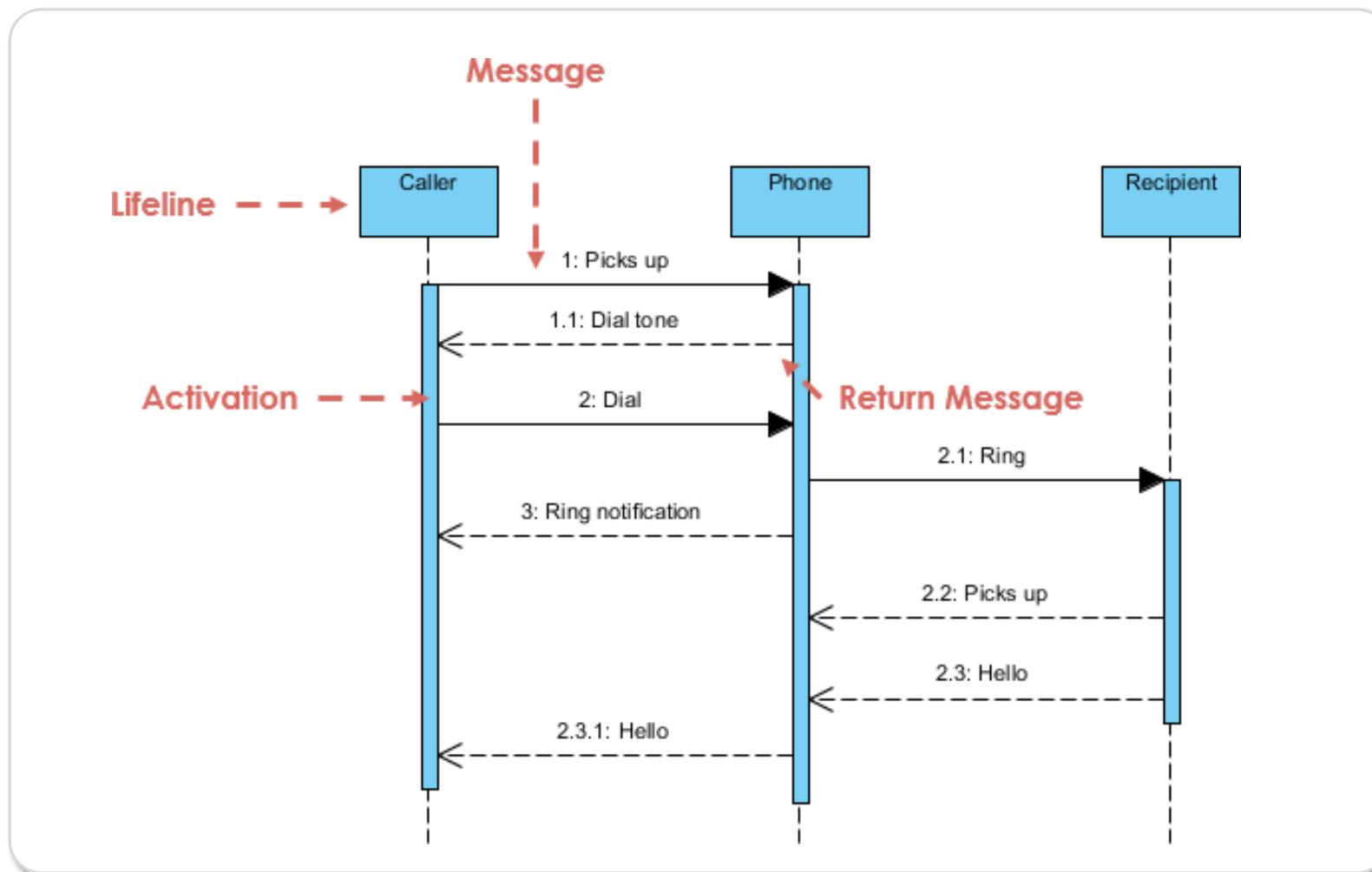
From Class Diagram
to
Object Diagram

Example of UML diagrams (*Behaviour*)



Use Case Diagram
(Banking System)

Example of UML diagrams (*Behaviour*)



Sequence Diagram
(Making a phone call)

Benefits of UML

1. เป็นภาษามาตรฐานหรือภาษาสากลที่ใช้ในการพัฒนาซอฟต์แวร์เชิงวัตถุ
2. ลดความซับซ้อนของระบบที่กำลังพัฒนาให้เห็นเป็นรูปธรรม
3. สามารถนำไปใช้ได้กับหลายๆ บริบท ไม่ใช่แค่กับการพัฒนาซอฟต์แวร์
4. ทำให้มีส่วนเกี่ยวข้องทุกคนเห็นภาพเดียวกัน และเข้าใจตรงกัน
5. ช่วยในการสื่อสารระหว่างทีมผู้พัฒนาระบบ
6. ช่วยให้เห็นข้อผิดพลาดที่สามารถเกิดขึ้นได้ หากทำให้เป็นโค๊ด

Activity Diagrams

Activity Diagrams

วัตถุประสงค์ของ Activity Diagrams

- ใช้ในการจำลองขบวนการ หรืองาน ในระบบงานธุรกิจ
- ใช้อธิบายพังก์ชันการทำงานของระบบ
- ใช้อธิบายตรรกะของการดำเนินการต่างๆ
- ในขบวนการพัฒนาซอฟต์แวร์โดยใช้ UML ใช้ Activity Diagram ใน การจำลองกิจกรรมต่างๆ

Notation of Activity Diagrams

Initial node

- ควรมีได้แค่ หนึ่ง initial node ต่อหนึ่ง activity diagram
- จุดดำทึบ

Decision node and Merge node

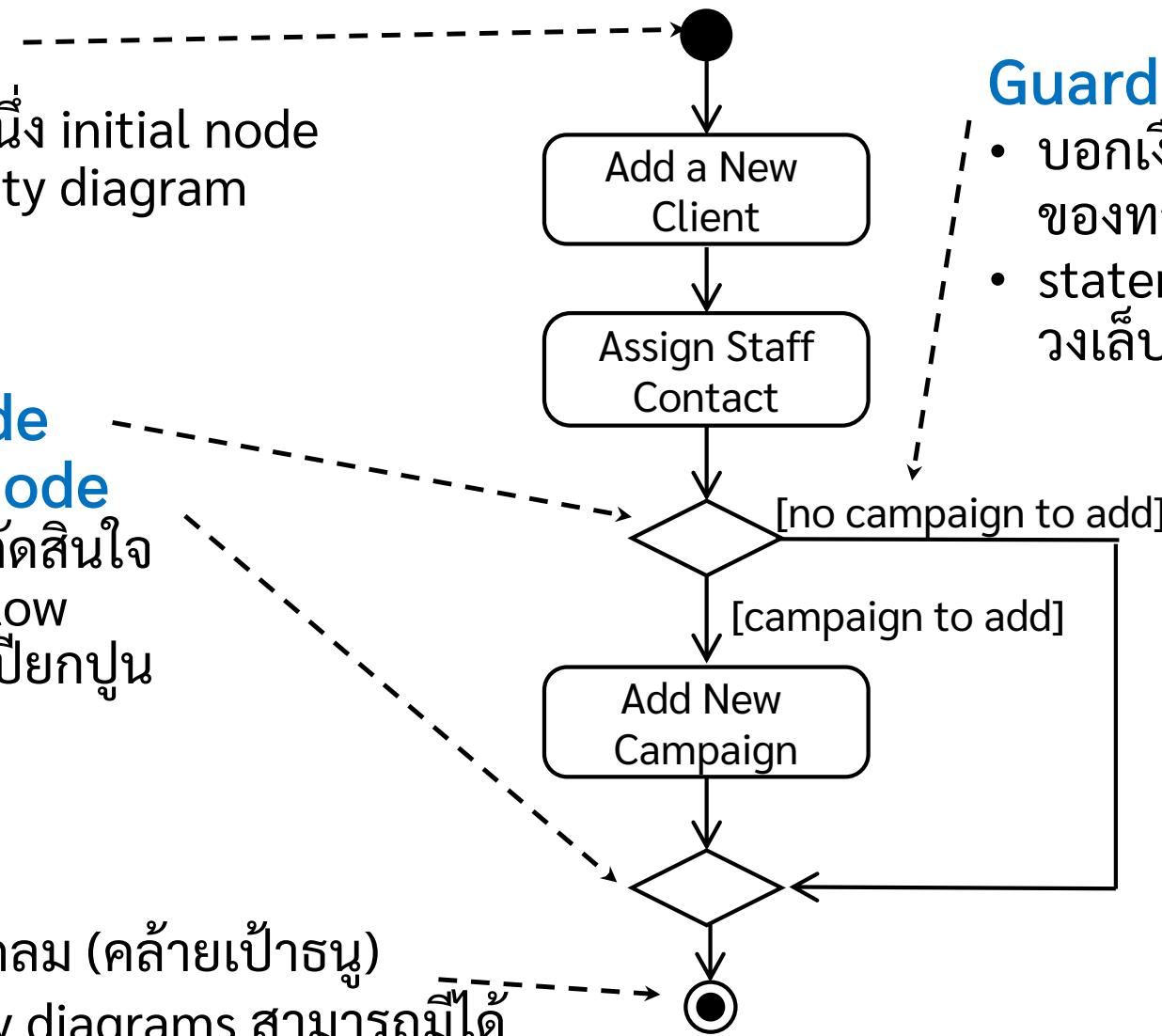
- ทางเลือกการตัดสินใจ และการรวม flow
- สีเหลี่ยมขนมเปียกปูน

Final node

- จุดดำทึบในวงกลม (คล้ายเป้าธนู)
- ในหนึ่ง activity diagrams สามารถมีได้มากกว่า หนึ่ง final node

Guard

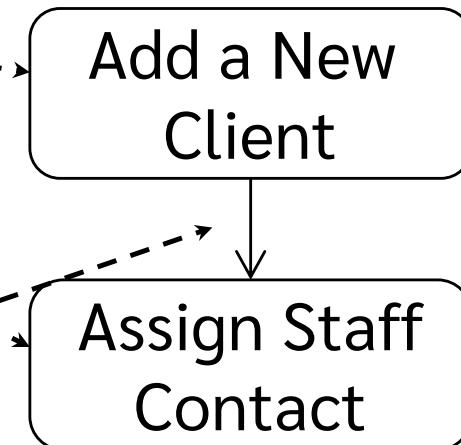
- บอกเงื่อนไขของทางเลือก
- statement ในวงเส้นสีเหลี่ยม



Notation of Activity Diagrams

- **Actions**

- rectangle with rounded corners
- meaningful name
- บอกถึงขั้นตอนต่างๆใน กิจกรรมหนึ่งๆ



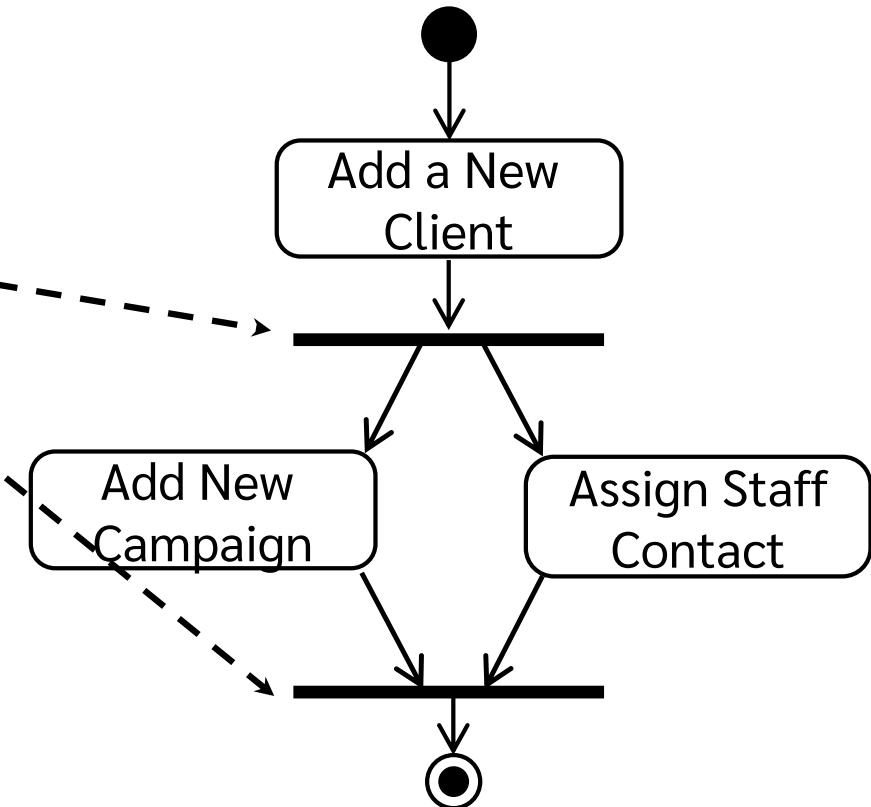
- **Control flow**

- arrows with open arrowheads
- ใช้วิธีดำเนินการเกิด activity ตามหัวลูกศร โดยจะเริ่มทำ action ถัดไปเมื่อ action ก่อนหน้าทำงานเสร็จ

Notation of Activity Diagrams

Fork node and Join node

- Actions carried out concurrently or in parallel
- ขั้นตอนบางขั้นตอนสามารถทำพร้อมกันได้ หรือแบบคู่ขนาน



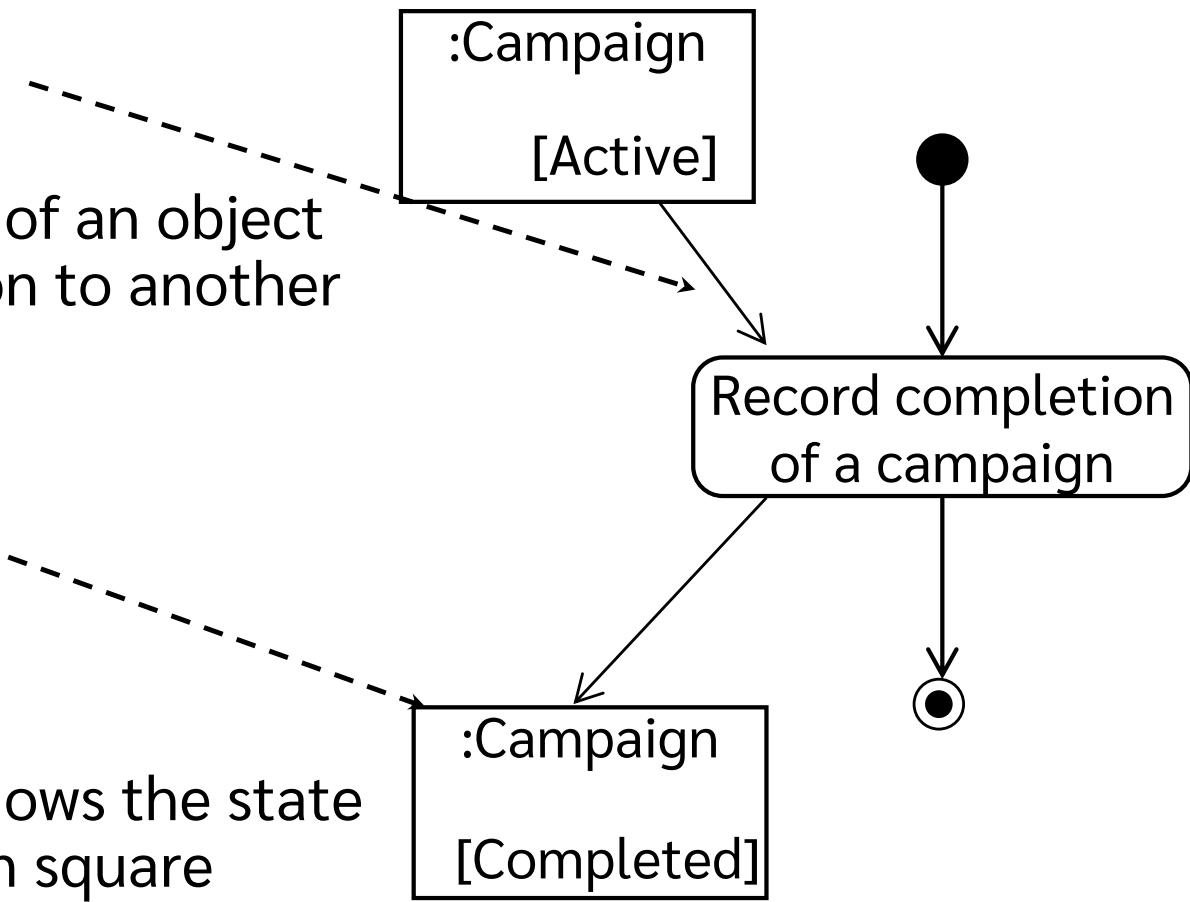
Notation of Activity Diagrams

- **Object flows**

- open arrow
- Show the flow of an object from one action to another action

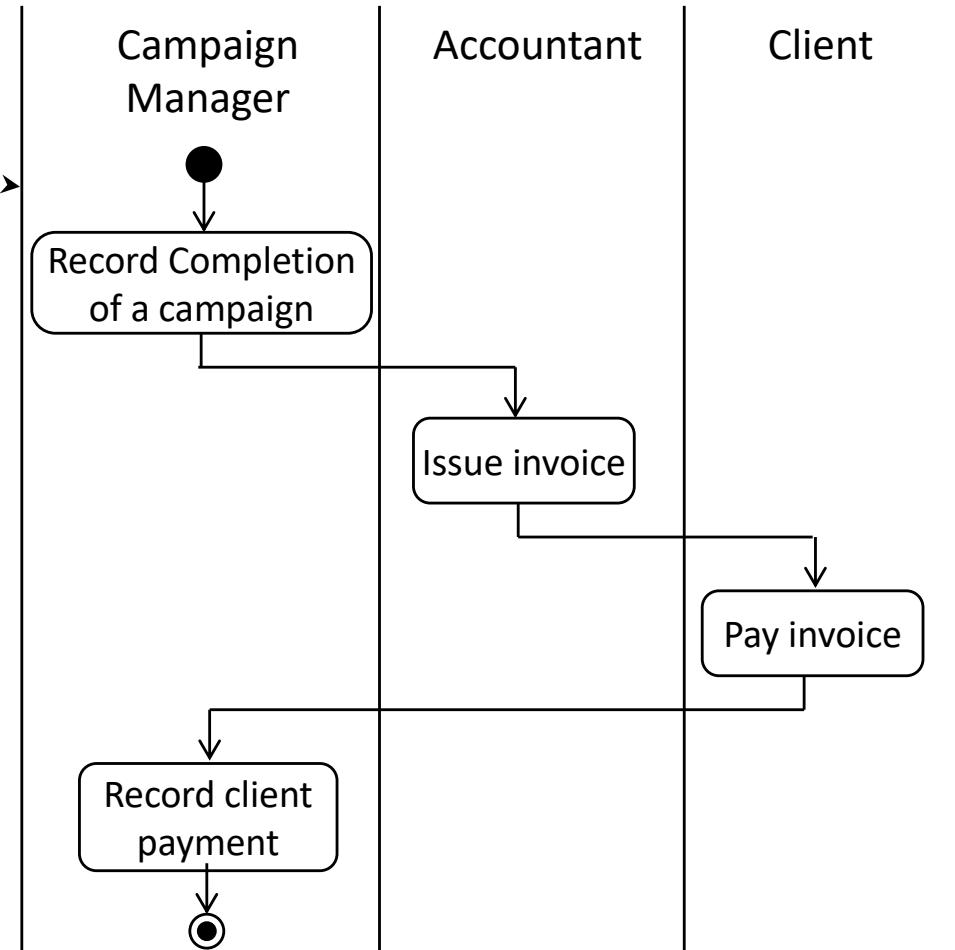
- **Objects**

- rectangle
- (Optionally) shows the state of the object in square brackets



Notation of Activity Diagrams

- **Activity Partitions (Swimlanes)**
 - vertical columns
 - คล้ายลู่ว่ายน้ำในสระ เลย เรียกว่า swimlanes
 - จะมีการกำหนดชื่อสำหรับเจ้าของ แสดงให้เห็นว่าใครเป็นผู้รับผิดชอบของขั้นตอนหนึ่งๆ ในกิจกรรมนี้

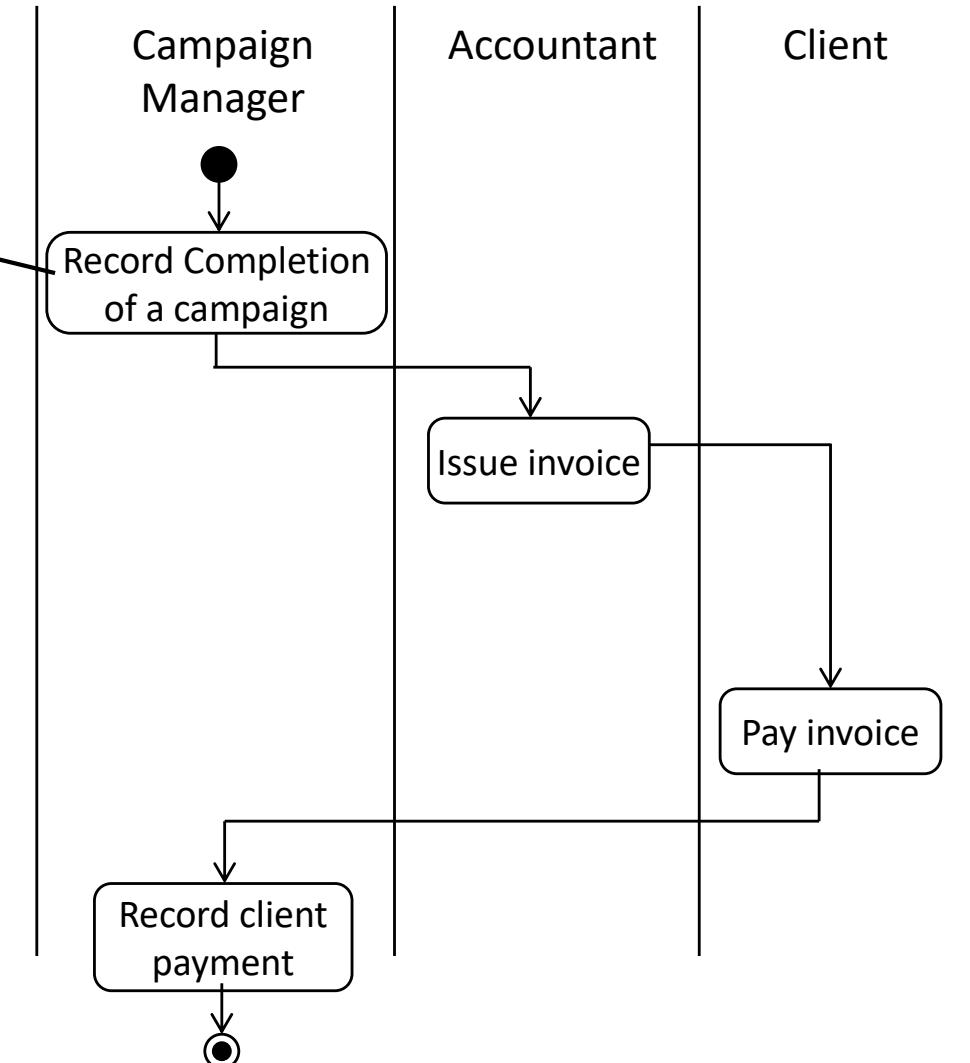


Notation of Activity Diagrams

Note

เราสามารถใส่ข้อความเพิ่มเติม
แยกได้ เพื่อเพิ่มข้อมูลและให้
ผู้อ่าน diagram เข้าใจมันได้ง่าย
ขึ้น

See Record
Completion of a
campaign diagram



Steps to Draw Activity Diagrams

ก่อนที่เราจะสร้าง Activity Diagrams เราต้องทราบ

- **วัตถุประสงค์ของมัน**

- จะทำให้เราเล็งเห็นว่าเราควรใส่ข้อมูลและมีรายละเอียดต่างๆ แค่ไหน

- **อะไรที่ต้องการใช้**

- ชื่อของกระบวนการธุรกิจ, use case, หรือการดำเนินงานต่างๆ

- **ความละเอียดของข้อมูลที่ต้องการใช้**

- เช่น ให้ครอ่าน ต้องการละเอียดมาก ละเอียดน้อย บางที่ action หนึ่งๆ สามารถแยกออกมาเป็นอีก activity diagram ออกมาได้อีก

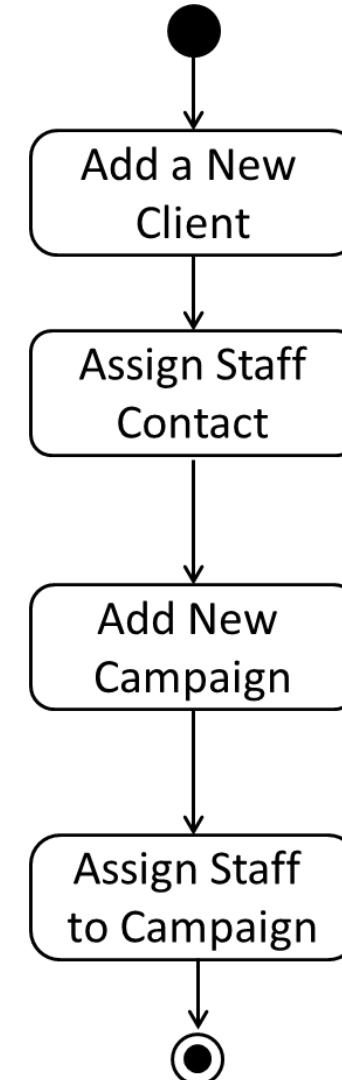
Steps to Draw Activity Diagrams

- **Identify Actions**

- (ระบุว่ามี actions อะไรบ้าง)
- เช่น
 - Add a new client
 - Assign staff contact
 - Add new campaign
 - Assign staff to campaign

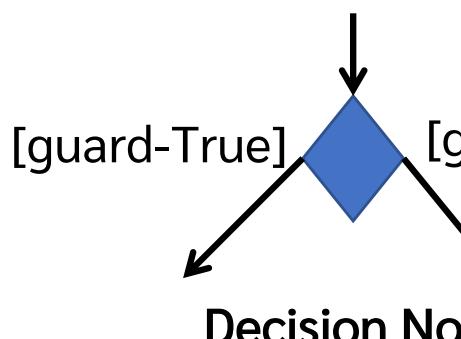
- **Organize the actions in order with flows (steps)**

- (จัดเรียง actions ตามการเกิดของมัน)

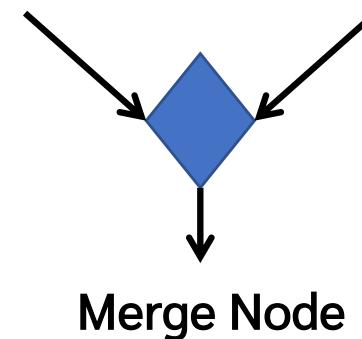


Steps to Draw Activity Diagrams

- Identify any alternative flows and the conditions on them.
 - หาว่า actions มีการกระทำเรามีทางเลือกอะไรหรือไม่ และเงื่อนไขต่างๆที่จะไปทางเลือกนั้นๆ
- Add decision and merge nodes, flows, and guard conditions to the diagram.
 - สร้าง decision nodes และ merge nodes มาเพื่อรับทางเลือกพวนน์
 - สร้าง guard conditions มากำหนดเงื่อนไขของทางเลือก



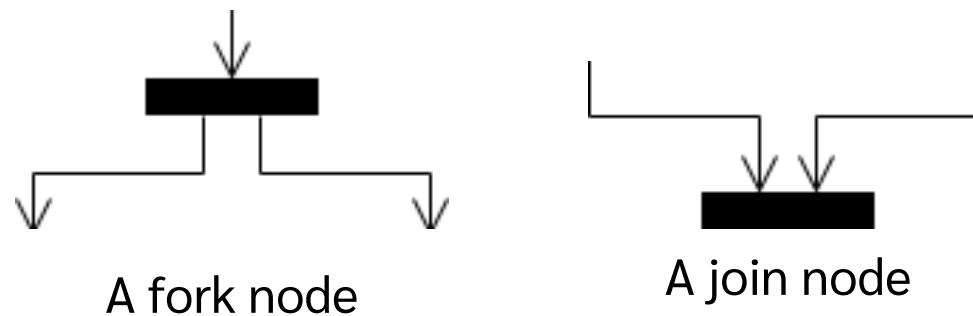
Decision Node



Merge Node

Steps to Draw Activity Diagrams

- Identify any actions that could be done concurrently or in parallel
 - หาว่ามี actions ไหนที่ทำร่วมกันหรือพร้อมกันได้
- Add fork nodes, join nodes, and flows, to the diagrams
 - สร้าง fork nodes และ join nodes เพื่อใช้ว่า actions เหล่านี้ สามารถทำพร้อมกันได้



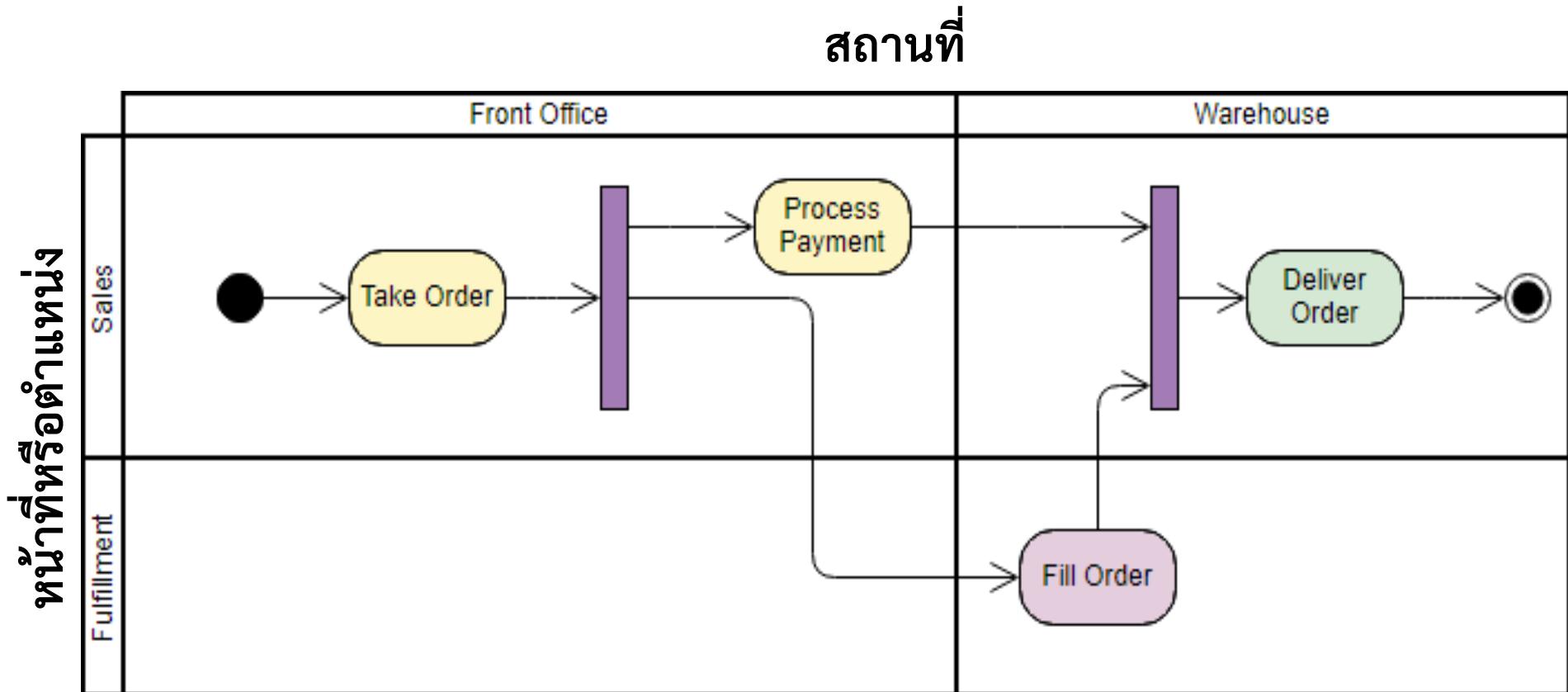
Steps to Draw Activity Diagrams

- Identify actors and group actions performed by the same actor
 - หาว่ามี actors มีใครบ้างและพยายามจัดรวมกลุ่ม actions ที่ทำโดย actor เดียวกันให้อยู่ด้วยกัน
- Add Swimlane or partition to the diagram
 - เพิ่ม swimlane หรือช่องแบ่งเพื่อให้เห็นว่ามี actors ไดบางและ actions ไดบางคน actor นั้นๆ



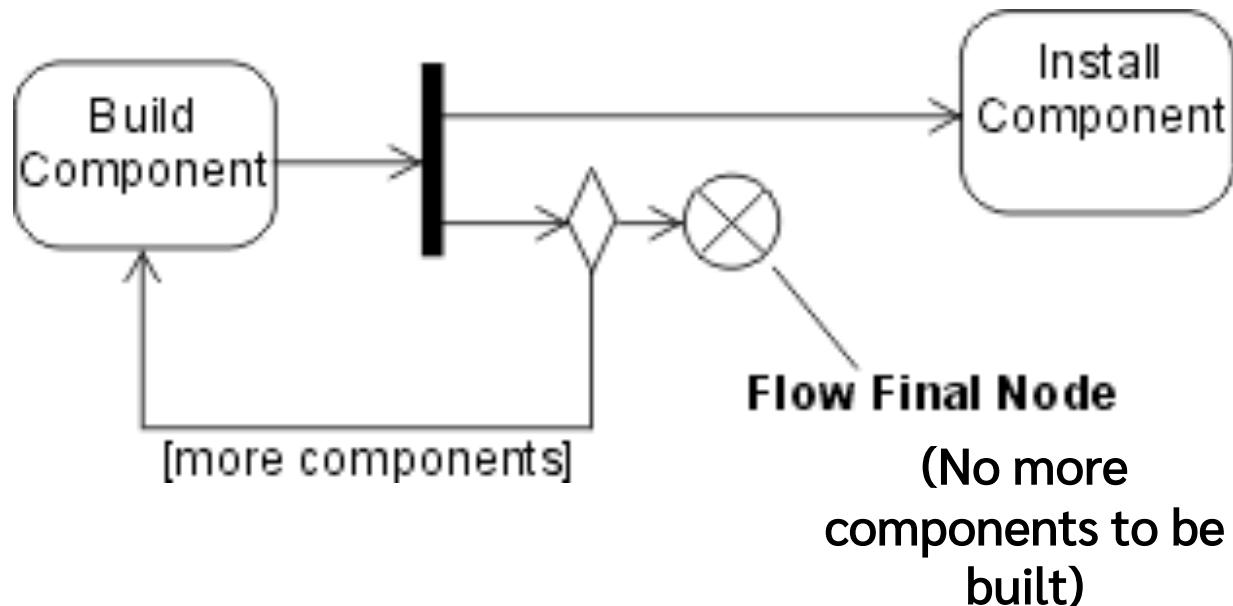
UML 2.0 - Activity Partitions

นอกจากจะแบ่ง Activity ในรูปแบบ Swimlane แล้ว ยังสามารถแบ่งกิจกรรมในมิติที่ 2 ได้ เช่น แบ่งตามสถานที่และหน้าที่หรือตำแหน่ง

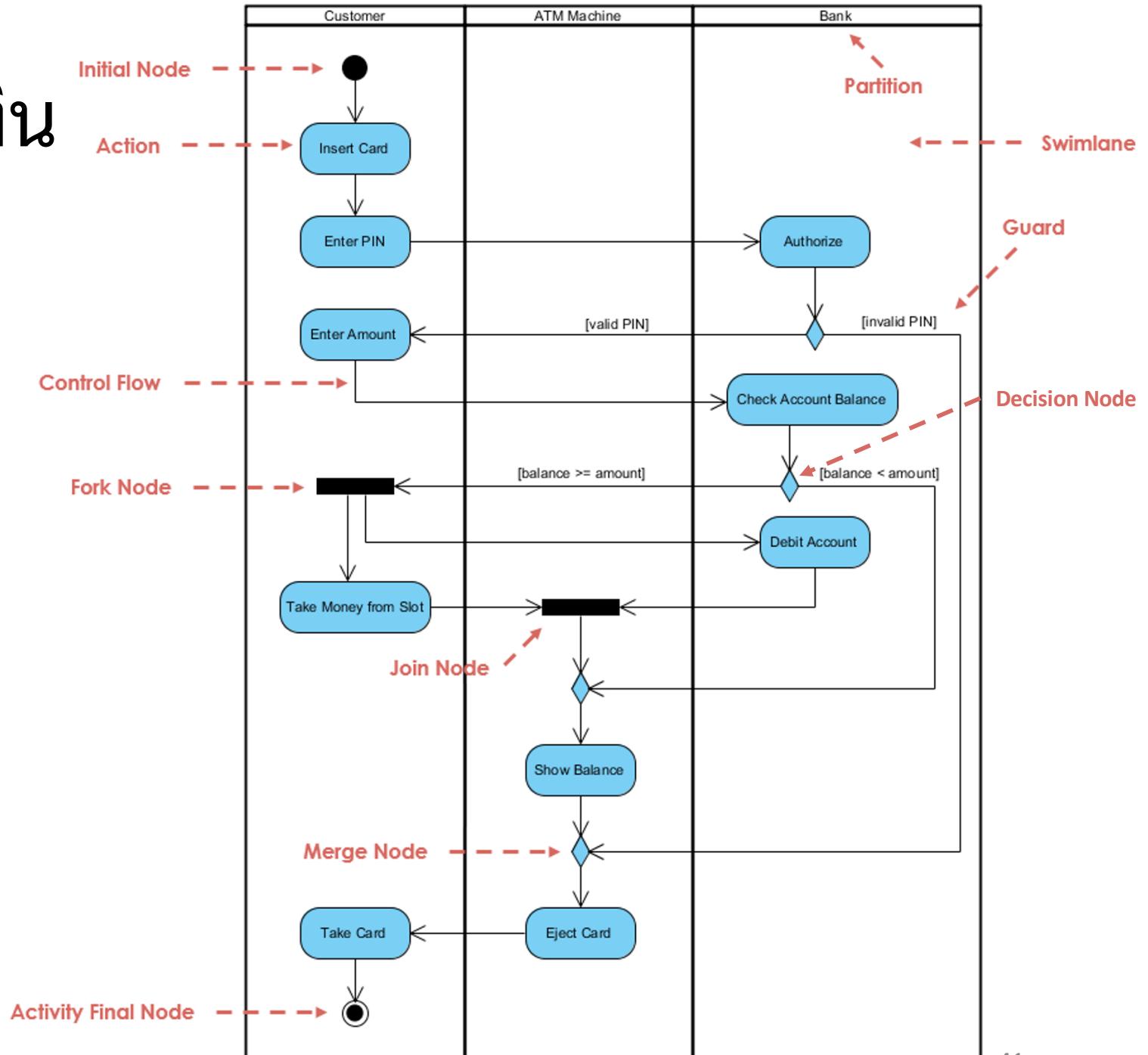


UML 2.0 – Flow Final

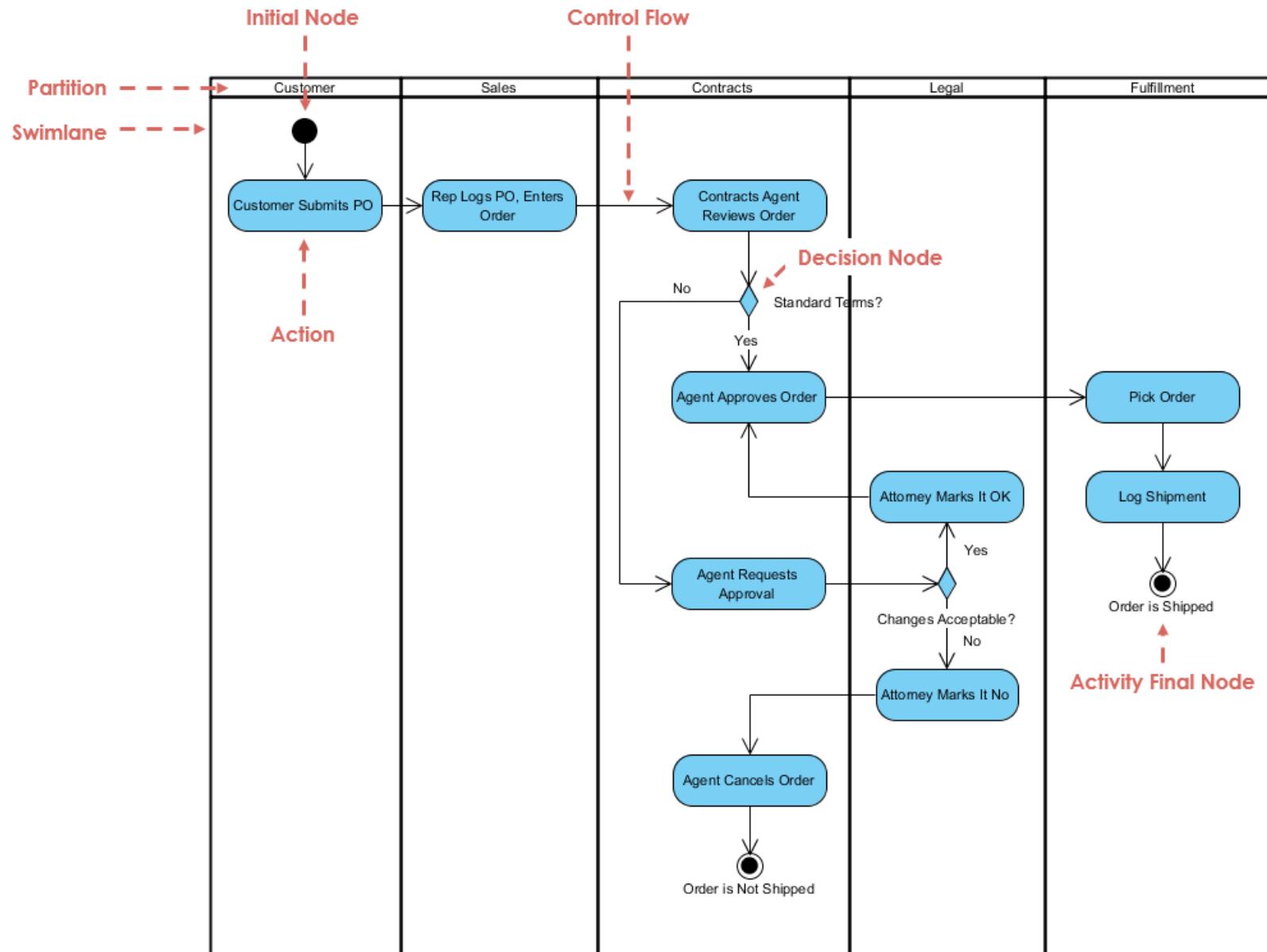
- **Flow Final:** An alternative to terminate a flow.
 - When control reaches any instance of Activity Final node, the entire activity (including all flows) is terminated. However, **only the flow that enters the Flow Final node exits the Activity.**



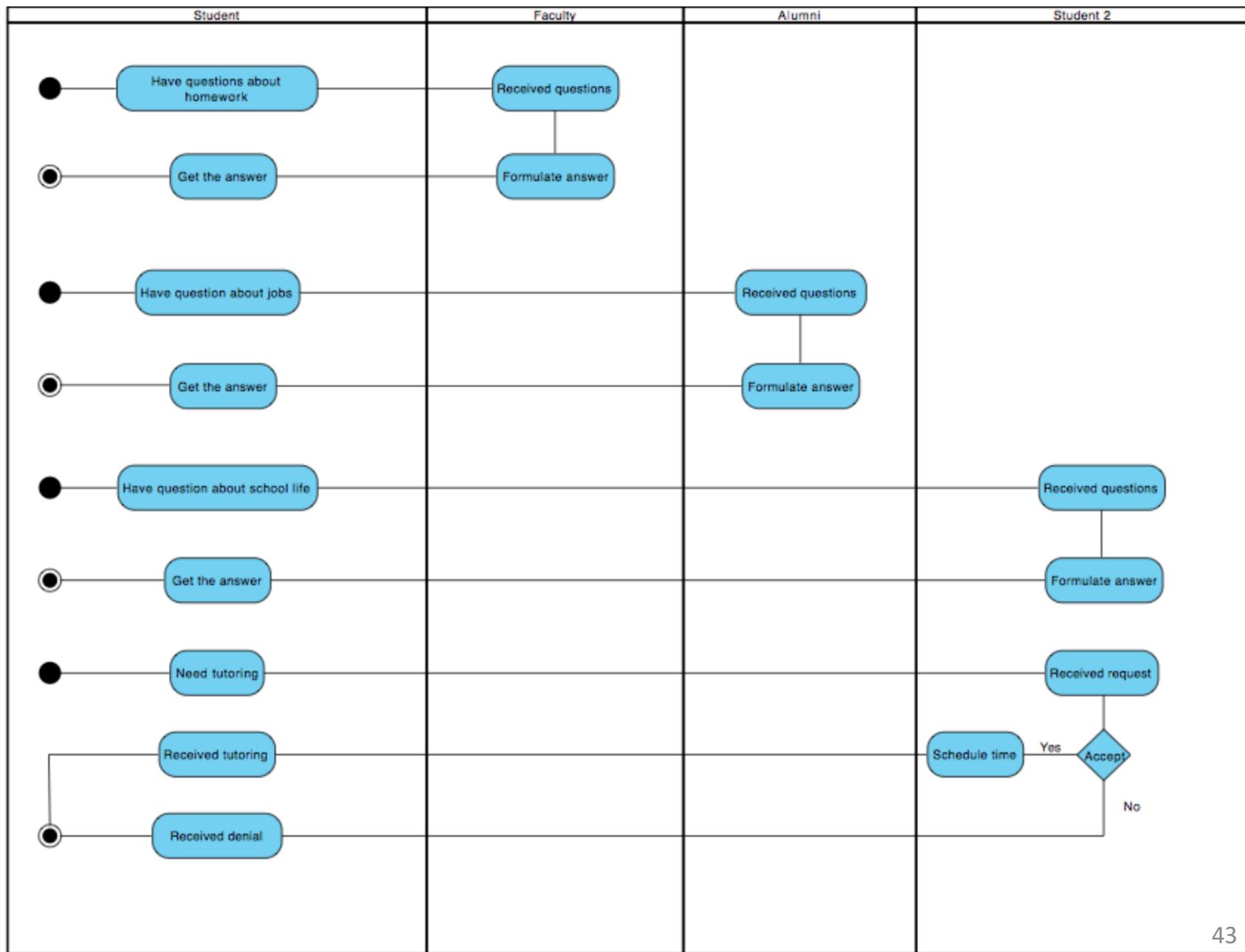
Example: ระบบถอนเงิน จาก ATM



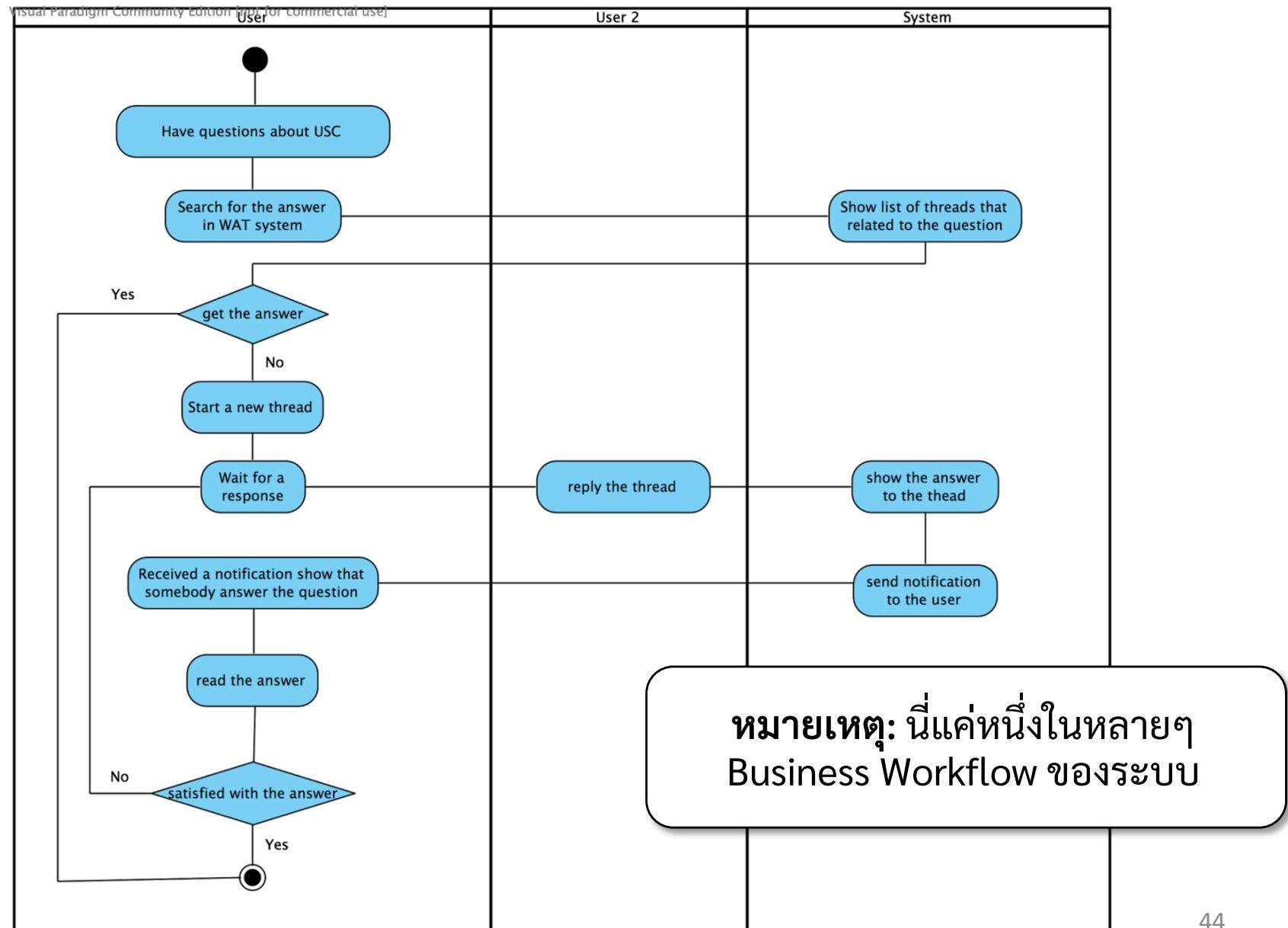
Example: Order Fulfillment



Example: We Are Trojans (Current)



Example: We Are Trojans (Proposed)



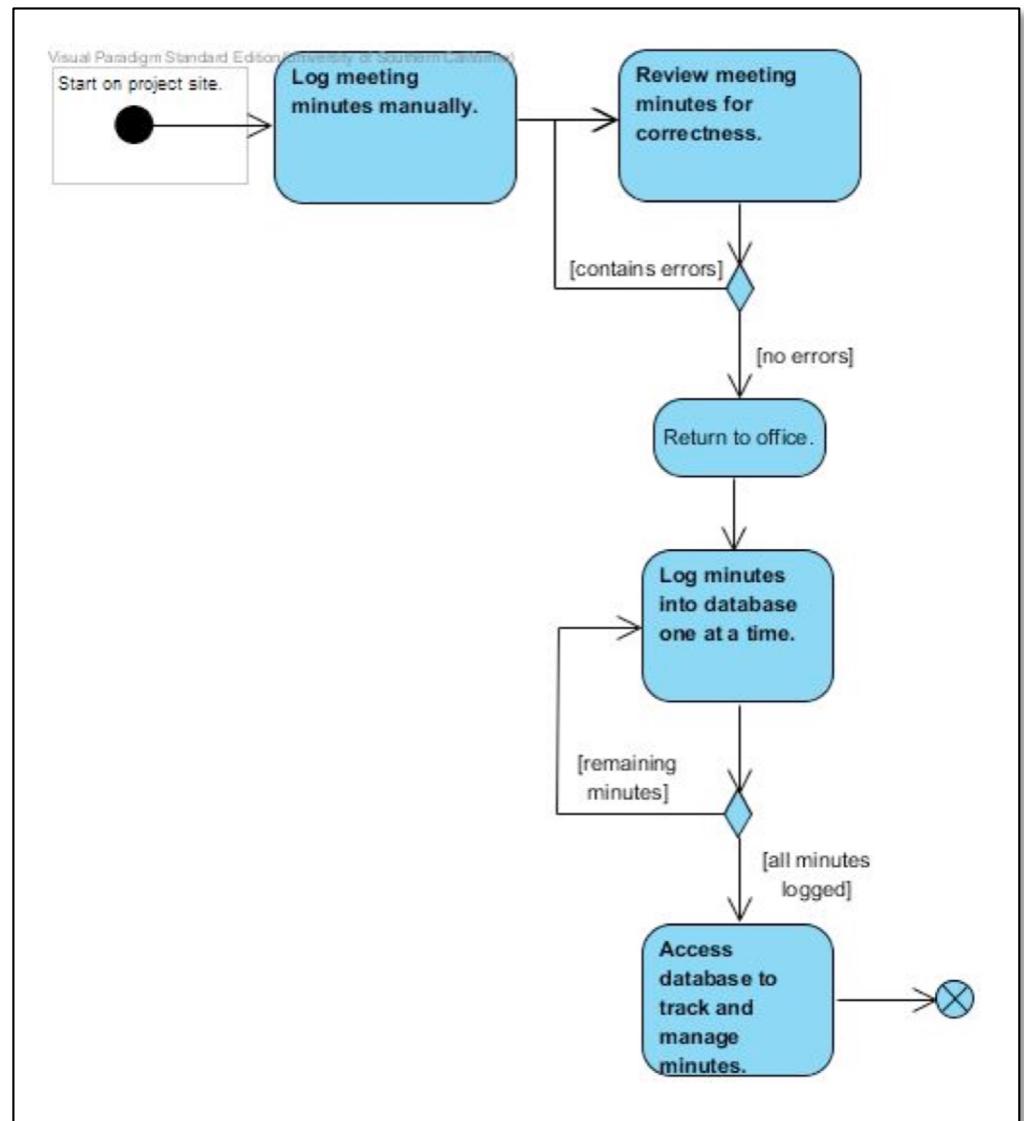
Example:

Project Description:

โปรเจคนี้ต้องการพัฒนา app ที่ เอาไว้ช่วยคนงานก่อสร้าง จดบันทึกการ หัวข้อและข้อมูลของประชุมผ่านทาง internet. ผู้ว่าจ้างบอกว่าปัจจุบันบริษัทใช้ การจดบันทึกการประชุมแบบ manual โดยให้พนักงานที่จดบันทึกเข้ามาส่ง รายงานการประชุมที่ office ใส่ข้อมูลลง ไปใน database จากเครื่องคอมที่ office



Current Business Workflow



Key Takeaways

- The distinction between models and diagrams.
- The purpose of activity diagrams
- The notation of activity diagrams
- How to draw activity diagrams

Questions?

