วิชา การวิเคราะห์และออกแบบเชิงวัตถุ

(Object Oriented Analysis and Design)

ผู้ช่วยศาสตราจารย์จุฑาวุฒิ จันทรมาลี
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสวนดุสิต

หน่วยที่ 5 โมเดลที่ใช้ออกแบบเชิงวัตถุ

สาระการเรียนรู้

1. ยูเคสไดอะแกรม
2. คลาสไดอะแกรม
3. ความสัมพันธ์ระหว่าง Class
4. ความสัมพันธ์ระหว่าง Object
5. หลักในการสร้าง Class Diagram
6. สเตตชาร์ตไดอะแกรม
7. การออกแบบสเตตไดอะแกรม
8. ซีเควนซ์ไดอะแกรม

สาระการเรียนรู้

9. ข้อสังเกตของ Sequence Diagram
10. คอลลาบอเรชั่นไดอะแกรม
11. แอกทิวิตี้ไดอะแกรม
12. ขั้นตอนในการเขียน Activity Diagram
13. คุณสมบัติของ Activity Diagram ที่ดี
14. คอมโพเนนต์ไดอะแกรม
15. ดีพลอยเมนต์ไดอะแกรม

จุดประสงค์การเรียนรู้

1. บอกความหมายของไดอะแกรมต่าง ๆ ได้
2. อธิบายความสัมพันธ์ระหว่าง Class และ Object ได้
3. อธิบายหลักในการสร้าง Class Diagram ได้
4. อธิบายวิธีการออกแบบสเตทไดอะแกรมได้
5. ประยุกต์ใช้โมเดลในการออกแบบเชิงวัตถุได้

สมรรถนะการเรียนรู้

1. แสดงความรู้เกี่ยวกับไดอะแกรมแบบต่าง ๆ
2. แสดงความรู้เกี่ยวกับหลักในการสร้าง Class Diagram และออกแบบสเตตไดอะแกรม
3. ปฏิบัติการเขียน Activity Diagram

ยูเคสไดอะแกรม (Use Case Diagram)

1. ส่วนประกอบสำคัญในยูสเคสไดอะแกรม มี 3 ส่วน

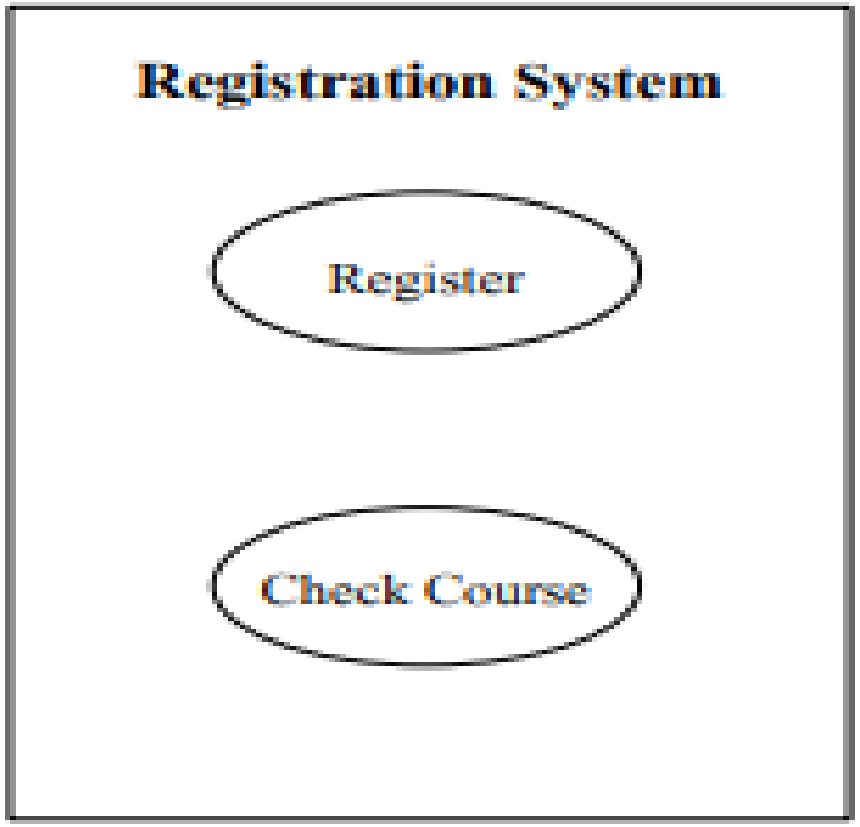
    1.1 ยูเคส (Use Case)

    1.2 แอ็กเตอร์ (Actor)

    1.3 เส้นแสดงความสัมพันธ์ (Relationship)

    ในการสร้างยูเคสไดอะแกรมสิ่งสำคัญคือการค้นหาว่าระบบ ทำอะไรได้บ้าง โดยไม่สนว่าจะทำงานอย่างไรหรือใช้เทคนิคการสร้างอย่างไร

ยูเคสไดอะแกรม (Use Case Diagram)



ยูเคสไดอะแกรม (Use Case Diagram)

2. ประโยชน์ของยูเคสไดอะแกรม สรุปได้ดังนี้

2.1 เพื่อให้ผู้พัฒนาทราบถึงความสามารถของระบบว่าต้องทำอะไรได้บ้าง

2.2 เพื่อทราบถึงผู้ใช้งานในแต่ละส่วนของระบบ

2.3 ทำให้การติดต่อสื่อสารระหว่างผู้พัฒนากับลูกค้าหรือระหว่างผู้พัฒนาด้วยกันทำได้ง่าย

2.4 ใช้ในการทดสอบระบบซอฟต์แวร์ ว่าทำงานได้ครบถ้วนตามความต้องการหรือไม่

คลาสไดอะแกรม (Class Diagram)

1. การสร้างคลาสไดอะแกรม
2. คำนามที่ปรากฏอยู่ในคำบรรยายยูเคสจะถูกสร้างเป็นคลาส

   เช่น คลาสรถยนต์ คลาสวิชาเรียน คลาสหนังสือ คลาสสินค้า
3. คำวิเศษณ์ที่ปรากฏอยู่ในคำบรรยายยูเคสจะถูกสร้างเป็นแอททริบิวท์

   เช่น สีรถ รุ่นรถ ยี่ห้อรถ
4. คำกิริยาที่ปรากฏอยู่ในคำบรรยายยูเคสจะถูกสร้างเป็นโอเปอเรชั่น

   เช่น สตาร์ตรถ เบรก ลงทะเบียน ยกเลิกรายวิชา
5. สัญลักษณ์

ความสัมพันธ์ระหว่าง Class

1. ความสัมพันธ์แบบพึ่งพา (Dependency)
2. ความสัมพันธ์แบบสืบทอดคุณสมบัติ (Inheritance)

   เช่น “Class แม่” (Super Class)

3. ความสัมพันธ์แบบร่วมกัน (Association)

ความสัมพันธ์ระหว่าง Object

1. Association เป็นความสัมพันธ์ระหว่าง Object หรือ Class แบบ 2 ทิศทาง

2. Aggregation เป็นความสัมพันธ์ระหว่าง Object หรือ Class แบบ “Whole-Part” หรือ “is part of” โดยจะมี Class ที่ใหญ่ที่สุดที่เป็น Object หลัก และมี Class อื่นเป็นส่วนประกอบ

3. Composition เป็นความสัมพันธ์ระหว่าง Object หรือ Class แบบขึ้นต่อกันและมีความเกี่ยวข้องกันเสมอ โดยจะมี Class ซึ่งเป็นองค์ประกอบของ Class อื่นที่ใหญ่กว่า เมื่อ Class ที่ใหญ่กว่าถูกทำลาย Class ที่เป็นองค์ประกอบก็จะถูกทำลายไปด้วย

ความสัมพันธ์ระหว่าง Object

4. Generalization เป็นความสัมพันธ์ระหว่าง Object หรือ Class ในลักษณะของการสืบทอดคุณสมบัติจาก Class หนึ่ง (Super class) ไปยังอีก Class หนึ่ง (Subclass)

5. Specialization คือ กระบวนการที่ตรงกันข้ามกับ กระบวนการ Generalization Abstraction กล่าวคือ ถ้าต้องการสร้าง Class ใหม่ โดยอาศัย Concept ของ Class เก่าบางส่วน และเพิ่มเติมใหม่บางส่วนจนเกิดเป็น Class ใหม่

หลักในการสร้าง Class Diagram

1. กำหนดกรอบของ Problem Domain ให้ชัดเจน
2. พิจารณาหา Objects ที่สามารถจับต้องได้เห็นได้สัมผัสได้ ซึ่งเรียกว่า Tangible Objects
3. พิจารณาหา Objects ที่ไม่สามารถจับต้องได้ซึ่งเรียกว่า Intangible Objects
4. ใช้ Classification Abstraction เพื่อแยกแยะและสร้าง Class จาก Objects ที่มีอยู่

หลักในการสร้าง Class Diagram

5. หา Aggregation Abstraction โดยพิจารณาการเป็นส่วนประกอบ
6. ใช้ Generalization มาพิจารณา
7. ใช้ Association มาพิจารณา
8. พิจารณา Class Diagram ว่ามี Class หรือ กลุ่มของ Class ที่ไม่มีความสัมพันธ์กับ Class อื่น ๆ หรือไม่

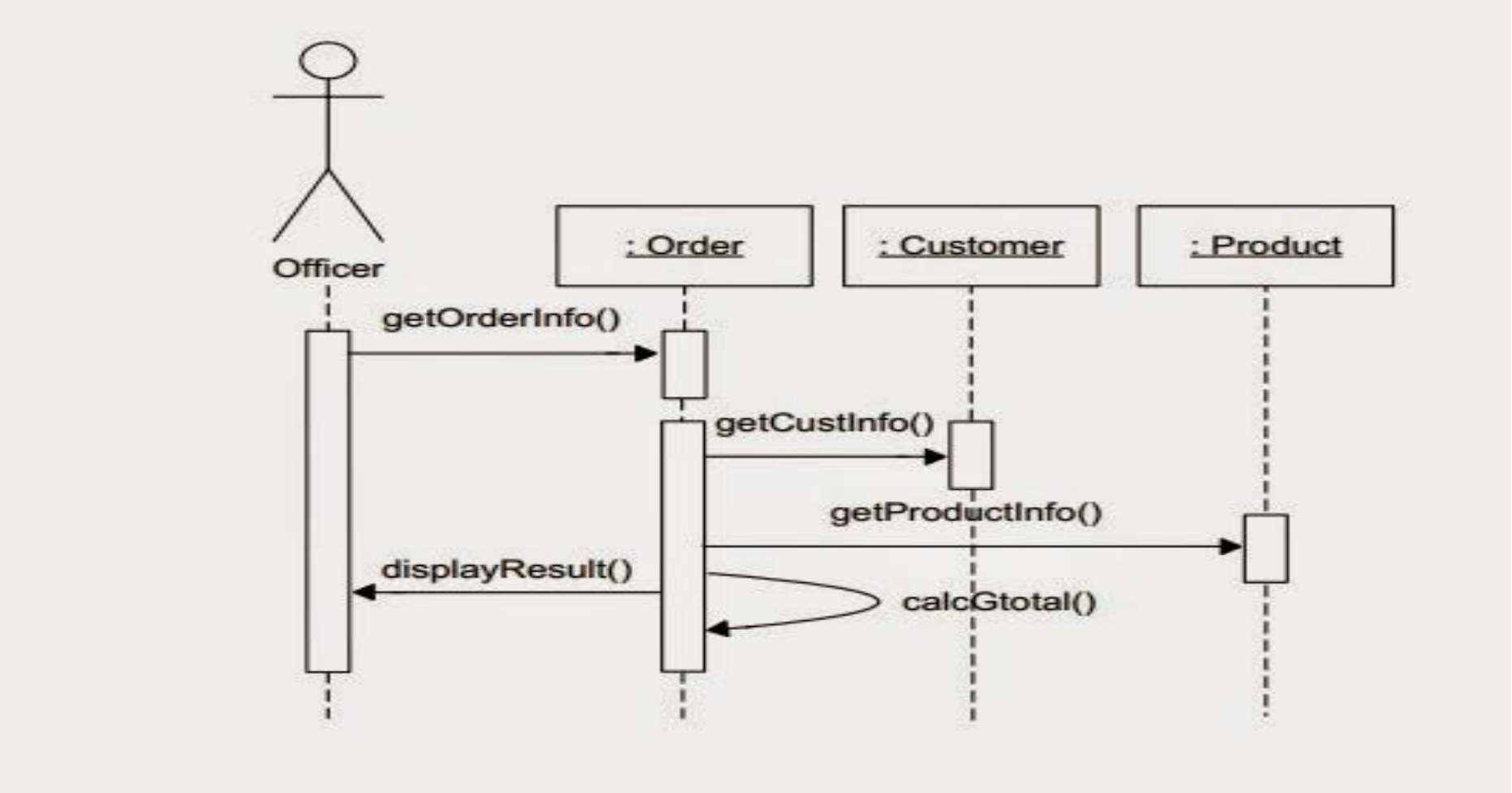
สเตตชาร์ตไดอะแกรม (State diagram)

สเตตชาร์ตไดอะแกรมบอกถึงพฤติกรรมของคลาสต่าง ๆ ในระบบว่ามีสถานะอะไรบ้างจะเปลี่ยน สถานะเมื่อเกิดเหตุการณ์อะไร สเตตชาร์ตไดอะแกรมของแต่ละคลาสประกอบไปด้วยสถานะที่สามารถเกิดขึ้น ได้ เช่น คนอยู่ในสถานะกำลังเดิน รถอยู่ในสถานะกำลังวิ่ง เป็นต้น เมื่อเวลาผ่านไปหรือมีเหตุการณ์บางอย่าง เกิดขึ้นย่อมทำให้เกิดการเปลี่ยนสถานะหรือเปลี่ยนพฤติกรรมได้

ซีเควนซ์ไดอะแกรม (Sequence Diagram)

ซีเควนซ์ไดอะแกรมบ่งบอกถึงในยูเคสนั้นวัตถุแต่ละตัวจะติดต่อสื่อสารกันอย่างไร มีขั้นตอนการทำงาน อย่างไร โดยเน้นไปที่แกนเวลาเป็นสำคัญถ้าเวลาเปลี่ยนขั้นตอนการทำงานจะเปลี่ยนโดยมีแอกเตอร์เป็น ผู้กระทำเริ่มต้น ในยูเอ็มแอลซีเควนซ์ไดอะแกรมมีแกนสมมติ 2 แกนคือ แกนนอนและแกนตั้ง

ซีเควนซ์ไดอะแกรม (Sequence Diagram)



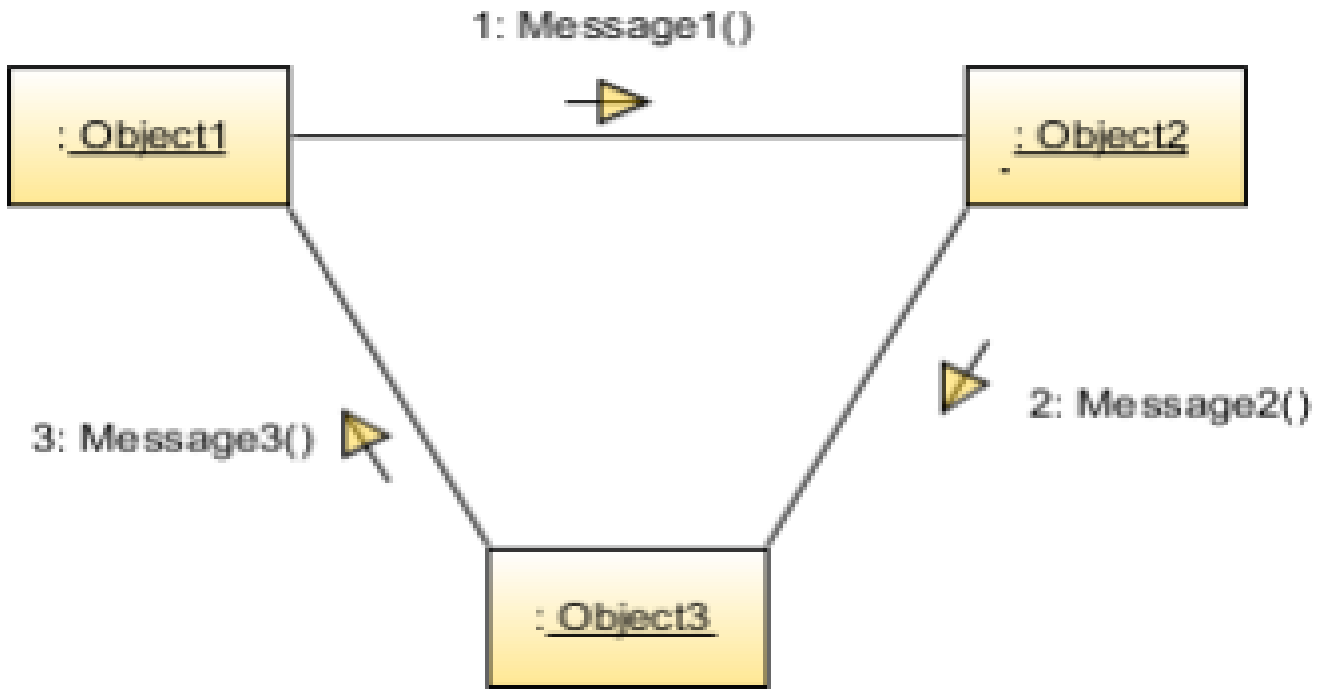
ข้อสังเกตของ Sequence Diagram

1. Sequence Diagram ช่วยให้นักวิเคราะห์ระบบทราบว่า คลาสใดควรจะมี Operation/Method ใดบ้าง
2. การเขียนลำดับกิจกรรมในแต่ละ Use Case สามารถเขียนได้อีกลักษณะหนึ่ง คือ แบ่งการปฏิบัติงานออกเป็น 2 ส่วน ได้แก่ ส่วนของ Actor ปฏิบัติ และส่วนของระบบปฏิบัติ

คอลลาบอเรชันไดอะแกรม (Collaboration Diagram)

มีหน้าที่เดียวกันกับซีเควนซ์ไดอะแกรมแต่ไม่แสดงถึงแกนเวลาอย่างชัดเจนยกเว้นการโต้ตอบกัน ระหว่างอ็อบเจกต์สัญลักษณ์ที่ใช้ประกอบด้วย วัตถุ หรือคลาสแทนด้วยรูปสี่เหลี่ยมคล้ายซีเควนซ์ไดอะแกรมมีรูปแบบคือ ชื่ออ็อบเจกต์/บทบาท : ชื่อคลาสและขีดเส้นใต้เพื่อแสดงว่าเป็นอินสแตนซ์ แต่ไม่จำเป็นต้องเรียง ตามแนวนอนเหมือนในซีเควนไดอะแกรมมีเส้นเชื่อมกันระหว่างวัตถุ เรียกว่า ลิงก์ (Link) ซึ่งแต่ละลิงค์มีคำอธิบายแสดงขั้นตอนการทำงานตามทิศทางลูกศรโดยมีตัวเลขลำดับกำกับไว้เพื่อบอกว่าขั้นตอนใดทำก่อนทำ หลังซึ่งแทนแกนเวลาตามด้วยเครื่องหมายทวิภาคและเมสเสจ

คอลลาบอเรชันไดอะแกรม (Collaboration Diagram)



แอ็กทิวิตี้ไดอะแกรม (Activity Diagram)

แอ็กทิวิตี้ไดอะแกรม (Activity Diagram) แอ็กทิวิตี้ไดอะแกรม แสดงลำดับกิจกรรมของการทำงาน (work Flow) โดยการแสดงทางเลือกที่เกิดขึ้นและขั้นตอนการทำงาน โดยประกอบไปด้วยสถานะต่าง ๆ ที่เกิดขึ้นระหว่าง การทำงาน และผลจากการทำงานในขั้นตอนต่าง ๆ วงกลมสีดำ คือ จุดเริ่มต้น เรียก Initial State วงกลมสีดำมีวงล้อมอีกชั้น คือ จุดสิ้นสุด เรียก Final State โดยจะอธิบายกิจกรรมในลักษณะของ การกระทำโดยใช้ไดอะแกรมใน UML จะมีลักษณะคล้าย Flow Chart

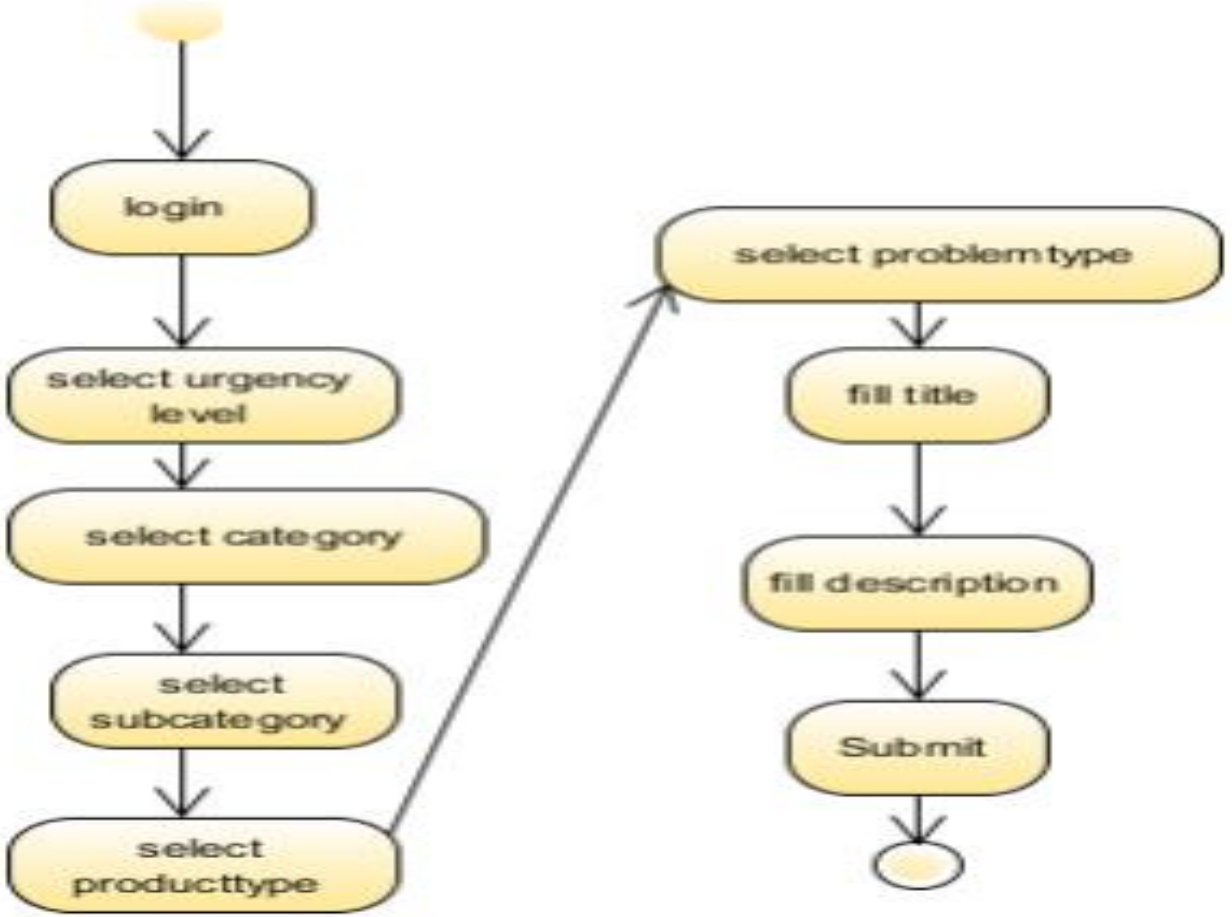
ขั้นตอนในการเขียน Activity Diagram

1. พิจารณากิจกรรมต่าง ๆ ที่ที่ได้จากผลการวิเคราะห์ที่ควรอธิบาย
2. พิจารณากิจกรรมย่อยที่เกิดขึ้น เงื่อนไขหรือกรณีต่าง ๆ ที่เกิดขึ้นเมื่อเป็นไปตามเงื่อนไข
3. เรียงลำดับกิจกรรมที่เกิดก่อนหลัง
4. เขียนกิจกรรมย่อยด้วยสัญลักษณ์แสดงกิจกรรม
5. เขียนจุดเริ่มต้น
6. เขียนจุดสิ้นสุด

คุณสมบัติของ Activity Diagram ที่ดี

1. มุ่งเน้นการติดต่อสื่อสารของระบบในเชิงไดนามิกส์
2. เฉพาะอิลิเมนต์ที่มีความสำคัญต่อกระบวนการทำงานเท่านั้น
3. แสดงรายละเอียดในแต่ละระดับการทำงาน โดยเลือกแสดง เฉพาะที่มีความสำคัญต่อการเข้าใจการทำงานของระบบเท่านั้น
4. ถ้าการทำงานส่วนใดมีความสำคัญ ก็ควรเขียน Activity Diagram ไม่ควรละเอาไว้หรือแสดงเพียงอย่างย่อ ๆ

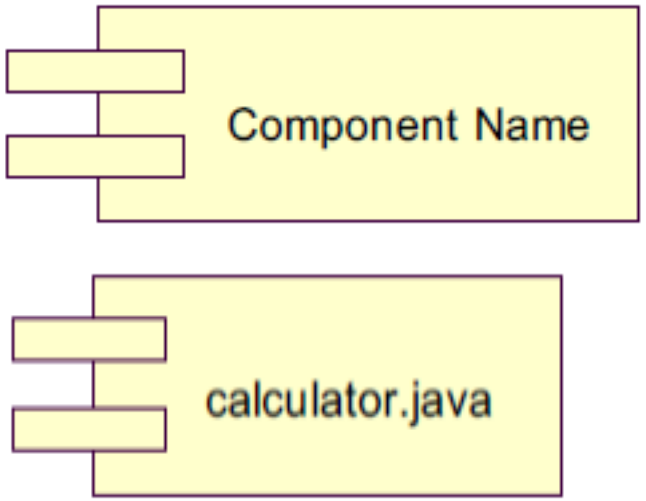
คุณสมบัติของ Activity Diagram ที่ดี



คอมโพเนนต์ไดอะแกรม Component Diagram)

คอมโพเนนต์ไดอะแกรม (Component Diagram) คอมโพเนนต์ไดอะแกรมเป็นไดอะแกรมที่แสดงโครงสร้างทางกายภาพของ Software โดยจะประกอบด้วยองค์ประกอบซึ่งอยู่ในรูปต่าง ๆ เช่น Binary, text และ Executable ภายใน Component Diagram ก็จะมีความสัมพันธ์แสดงอยู่เช่นเดียวกับ Class Diagram, Object Diagram เป็นไดอะแกรมที่แสดงโครงสร้างและความเกี่ยวข้องกันของซอฟต์แวร์โดยคอมโพเนนต์ ประกอบไปด้วย source code และ runtime หรือ executable component

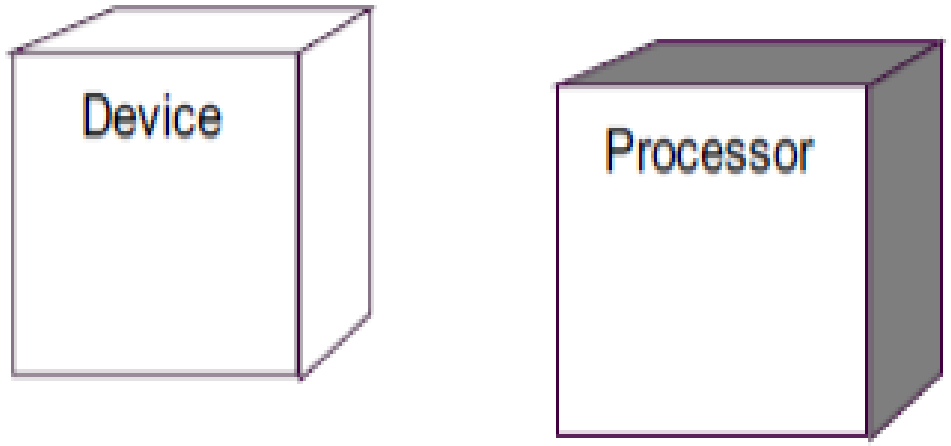
คอมโพเนนต์ไดอะแกรม Component Diagram)

- 

ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram)

แสดงการเชื่อมต่อของอุปกรณ์ฮาร์ดแวร์ในระบบและมักใช้ร่วมกับคอมโพเนนต์ไดอะแกรมโดยข้างใน ฮาร์ดแวร์อาจประกอบไปด้วยซอฟต์แวร์คอมโพเนนต์ดีพลอยเมนต์ไดอะแกรมแสดงอยู่ในรูปอินสแทนซ์ และ แสดงในช่วงเวลาของการรันหรือระหว่างการเอ็กซิคิวต์ ดังนั้นไฟล์คอมโพเนนต์ของระบบที่ไม่ได้ใช้สำหรับรันจะไม่ปรากฏในไดอะแกรมนี้แต่มีในคอมโพเนนต์ของไฟล์ที่ใช้ทำงานจริงเท่านั้น

ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram)



จบการนำเสนอ