**บทที่ 6**

**แบบจำลองปฏิสัมพันธ์ (Interaction Modeling)**

**จุดประสงค์**

1. เพื่อให้นิสิตข้าใจถึงแผนภาพที่แสดงถึงลำดับขั้นตอนการทำงานภายในระบบ
2. เพื่อให้นิสิตข้าใจถึงแผนภาพ Sequence Diagramและ Collaboration Diagram

**แบบจำลองปฏิสัมพันธ์**

แผนภาพยูสเคสเป็นแบบจำลองในกลุ่ม Dynamic หรือ Behavior View (มุมมองเชิงพฤติกรรม) ใช้แสดงหน้าที่ของระบบ ซึ่งเก็บจากความต้องการของผู้ใช้ แต่ไม่แสดงลำดับขั้นตอนการทำงานภายใน

แผนภาพคลาสเป็นแบบจำลองในลักษณะ Static หรือ Structure View (มุมมองเชิงโครงสร้าง) โดยจะแสดงโครงสร้างของระบบที่ไม่มีการเคลื่อนไหว ไม่ระบุขั้นตอนการดำเนินงาน ลำดับการทำงานก่อนหลัง

ในบทนี้จะกล่าวถึงแบบจำลองในมุมมอง Dynamic หรือ Behavior View ที่แสดงถึงลำดับขั้นตอนการทำงานภายในระบบ ซึ่งแบบจำลองกลุ่มนั้น คือ Interaction Modeling

ซึ่งจะประกอบด้วยแผนภาพย่อยSequence Diagram และ Collaboration Diagram

ทั้ง 2 แผนภาพ แสดง Messages ที่ถูกส่งผ่านระหว่าง Objects ที่ทำงานร่วมกัน เพื่อประกอบเป็นหน้าที่การทำงานของระบบ

Sequence diagrams เน้น Message ที่เกิดขึ้นตามลำดับของเวลา

Collaboration diagrams เน้นการเชื่อมต่อ(Link)ทางด้านโครงสร้าง ระหว่างวัตถุที่ทำงานร่วมกัน

**แบบจำลองปฏิสัมพันธ์**

* + อธิบายกิจกรรมที่เกิดขึ้นในระบบ
  + แสดงปฏิสัมพันธ์ (Interaction) ระหว่างกลุ่มของ Objects
  + อธิบายการติดต่อสื่อสารระหว่าง Objects
  + ปกติจะใช้ 1 Interaction Diagram เพื่ออธิบายกิจกรรมใน 1 Use Case

**องค์ประกอบสำคัญที่ใช้ใน Interaction Diagram**

1. Objects Objectที่เป็นผู้ส่ง Message เรียกว่า Sender Object ที่รับ Message และดำเนินการตาม Message เรียกว่า Receiverซึ่ง Objects ใน Interaction Diagram แบ่งได้เป็น

**Concrete Object** คือ Object ที่สามารถระบุตัวตนได้ (Unique Identity) ของ Object นั้นได้เช่น นายสมชาย เกิดดี

**Prototypical Object** คือ Object ที่ไม่สามารถระบุตัวตนได้ (Unique Identity) ของ

Object นั้นได้อย่างแน่นอน เช่น Object ของ Class Student

การใช้สัญลักษณ์ของ Object ใน Interaction Diagram

**Concrete Object**

ชื่อ Object **:** ชื่อ Class

**Prototypical Object**

ชื่อ Class

**** ซึ่งจะระบุใน

ตัวอย่าง

**Concrete Object Prototypical Object**

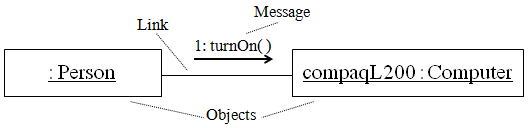


**2. Links**

เป็นสื่อกลางหรือทางเดินของ Message ที่รับ/ส่งกันระหว่าง 2 Object (เชื่อมโยง Object เข้าด้วยกัน)

ถ้า Class มีความสัมพันธ์ (Association) ระหว่างกันแล้วObject ของ Class นั้นก็จะมี Link ระหว่างกันด้วย

**ตัวอย่าง** Interaction Diagram(Collaboration Diagram) ที่มี Link ระหว่าง Object



**3. Message**

เป็นการสื่อสารระหว่าง 2 Object

ส่วนใหญ่จะเป็นการที่ Sender Object เรียกใช้(Call) Method ของ Receiver Object

Message แบ่งได้เป็น 5 ประเภทดังนี้

**Call หรือ Simple**

Message ที่ไม่ระบุรายละเอียดของวิธีการติดต่อสื่อสารระหว่าง Objects

 ใช้สัญลักษณ์

Return

Message ที่ใช้ส่งข้อมูลจาก Receiver กลับไปยัง Sender

(หลังจากที่ Sender ร้องขอ)

 ใช้สัญลักษณ์

Synchronous

Message สำหรับเรียกใช้ Method ของ Object

โดย Sender รอจนสิ้นสุด Method

 ใช้สัญลักษณ์

**Asynchronous**

Message สำหรับเรียกใช้ Method ของ Object

 โดย Sender ทำงานต่อทันทีโดยไม่รอจนสิ้นสุด Method ใช้สัญลักษณ์

 Create Message ที่ Object ตัวหนึ่งส่งออกไปเพื่อสร้าง Object ขึ้นจาก Class ใดๆ

Destroy Message ที่ Object ตัวหนึ่งส่งไปยังอีก Object หนึ่งเพื่อให้ Object ที่ได้รับ Message นั้น

 ทำลายตัวเอง

**การกำหนดลำดับ(Sequencing)**

**Sequencing** เป็นกลไก ที่ใช้ในการกำหนดลำดับการเกิดขึ้นของ Message ซึ่งเป็นสิ่งจำเป็น เพราะกิจกรรมของระบบเกิดขึ้นจากการส่งผ่าน Messages ระหว่าง Objects ต่างๆในระบบ

ตัวอย่าง Interaction Diagram (Collaboration Diagram) ที่มีการระบุ Sequencing

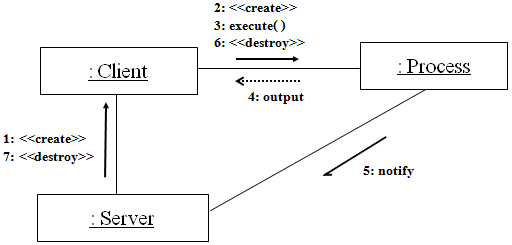


Diagram ในกลุ่ม Interaction Diagram ใน UML คือ Sequence Diagram และ Collaboration Diagram ซึ่งทั้ง 2 Diagram ใช้แสดงปฏิสัมพันธ์ (Interaction) ระหว่าง Object ตามลำดับขั้นตอนของเหตุการณ์เหมือนกัน

Sequence Diagram (แผนภาพแสดงลำดับเหตุการณ์) เน้นแสดงเวลาเป็นหลัก

Collaboration Diagram เน้นแสดงความร่วมมือของ Object อย่างชัดเจน ปกติจะเลือกใช้เพียง Diagram ใด Diagram หนึ่ง ใน Interaction Diagram (Sequence และ Collaboration Diagram)

อาจประกอบด้วย

**Messages with parameters**

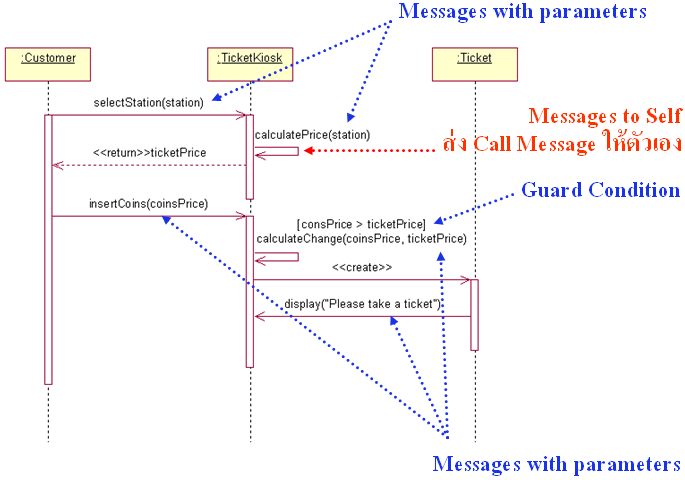
สามารถส่งผ่าน parameters ไปพร้อมกับ Message ได้ โดยเฉพาะการเรียกใช้ Methods ที่ต้องการ parameters ของ Object อื่น (Call Message)

**Guard Condition**

ในกรณีที่ Message ใดๆ เกิดขึ้นอย่างมีเงื่อนไข สามารถระบุเงื่อนไขที่ทำให้เกิด Message ไว้ข้างหน้าชื่อ Message ในรูปแบบ

[Guard Condition]Message Name

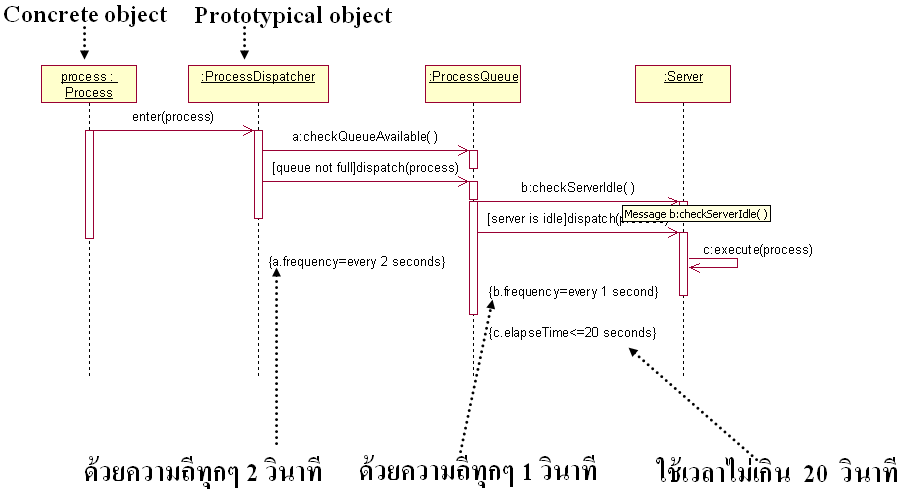
โดยที่ ถ้าเงื่อนไขเป็นจริง จึงจะมีการส่ง Message

ตัวอย่าง Sequence Diagram ที่ประกอบด้วย **Messages with parameters และ Guard Condition**

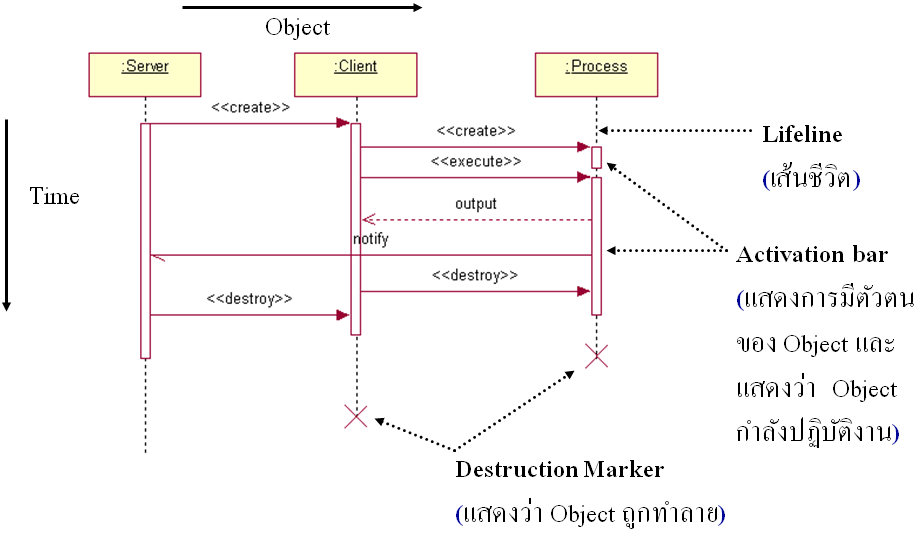
**Timing Constraints Modeling**

Timing Constraintsเป็นสิ่งที่ใช้ในการกำหนดเวลาที่ใช้ดำเนินการตาม Message และความถี่ของ Message

**ตัวอย่าง** Sequence Diagram ที่มีการอธิบาย Timing Constraints



**Sequence Diagram** หรือ **แผนภาพแสดงลำดับเหตุการณ์**เป็น Diagram หนึ่งในกลุ่มของ Interaction DiagramSequence Diagram มุ่งเน้นอธิบายกิจกรรม หรือ ปฏิสัมพันธ์(Interaction) ระหว่าง Object ตามลำดับของเวลา (Time-Ordering Description) หรือ ลำดับก่อนหลังของการเกิด Message Sequence Diagram ไม่สนใจความสัมพันธ์ระหว่าง Objects ( ไม่มี Link เหมือนกับ Collaboration Diagram )



**ประโยชน์ของ Sequence Diagram**

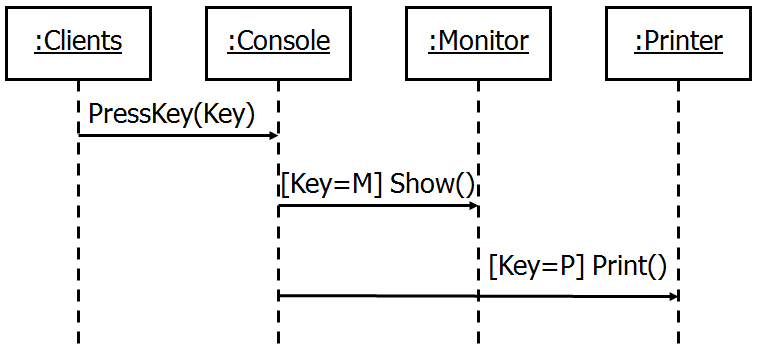
* + ช่วยแสดงรายละเอียดของลำดับขั้นตอนในการทำงานของแต่ละ Use Case ทั้งหมดที่มีอยู่ในระบบ ( Use Case เป็นการวิเคราะห์ระบบในภาพรวม )
  + ช่วยในการพิจารณาว่า Class ใน Class Diagram ที่สร้างขึ้นมีการทำงาน (Method) ใดขาดหายไป หรือควรเพิ่มเติมเข้าไปอีกหรือไม่

ซึ่งจะทำให้ Class ต่าง ๆ ที่สร้างขึ้นมีความสมบูรณ์มากยิ่งขึ้นClass Diagram เบื้องต้นที่สร้างขึ้นในบทก่อนจะยังไม่ใส่รายละเอียด Method ลงไป ซึ่งจะใส่ Method ลงไปในขั้นตอนนี้

**ขั้นตอนการสร้าง Sequence Diagram**

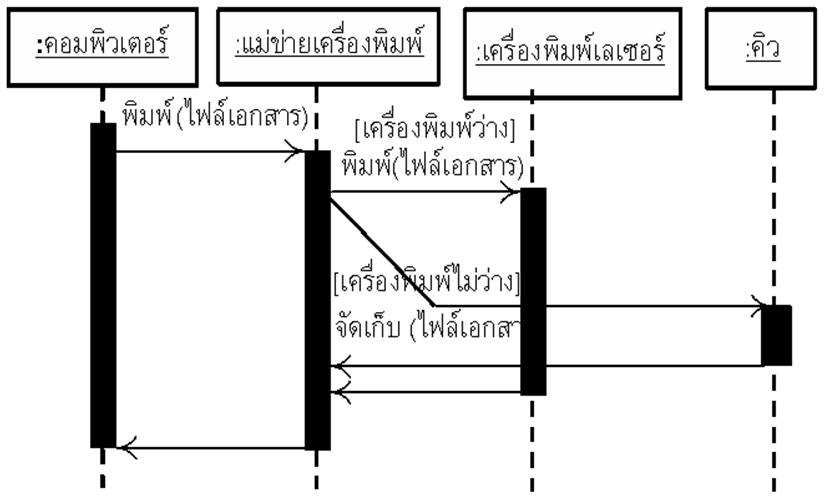
* + พิจารณาทีละ Use Case โดยยังไม่ต้องคำนึงถึงความสัมพันธ์ที่แต่ละ Use Case มีต่อกัน
  + พิจารณาแต่ละ Use Case ว่ามี Class หรือ Objects ใดร่วมทำให้เกิด กิจกรรมใน Use Case นั้น ๆ บ้าง
  + นำเอา Class หรือ Object ต่าง ๆ มาเรียงต่อกันในแนวนอนโดยให้นำ Actor ไว้ที่ด้านซ้ายสุดเสมอ แล้วนำเอา Class หรือ Object ต่างๆ เรียงต่อกันจากซ้ายไปขวา
  + หาก Use Case นั้นมี Actor โดยปกติกิจกรรมแรกที่ถูกเรียกมักจะเกิดจาก Actor ก่อนเสมอ ดังนั้นเมื่อเกิดกิจกรรมไปที่ Class หรือ Object ใด ให้ย้าย Class หรือ Object นั้นมาทางซ้าย ทำเช่นนี้เรื่อย ๆ จนกระทั่ง กิจกรรมทั้งหมดครบถ้วน
  + กรณีที่มีกิจกรรมเกิดขึ้นใหม่ แต่กิจกรรมที่เกิดขึ้นนั้นไม่มีใน Class หรือ Object ที่ลูกศรชี้ไป ให้เข้าไปเพิ่มกิจกรรม (Method) นั้น ๆ ลงไปที่ Class นั้นใน Class Diagram
  + หากต้องมีการเพิ่ม Class ใหม่เข้าไปใน Sequence diagram ต้องเข้าไปเพิ่มเติม Class นั้นและ Relationship ที่มีทั้งหมดใน Class Diagram ด้วย
  + ทำจนครบทุก Use Case
  + นำ Class และกิจกรรมที่เกิดขึ้นใน Use Case ที่ถูก Includeแทรกเข้าไปใน Use Case ที่เรียกใช้ และใช้กิจกรรมเพื่อเชื่อมโยง Sequence Diagram ทั้งสอง
  + นำ Class และกิจกรรมที่เกิดขึ้นใน Use Case ที่ถูก extend แทรกเข้าไปใน Use Case ที่ถูก extend
  + และใช้กิจกรรมเพื่อเชื่อมโยง Sequence Diagram ทั้งสอ

**ตัวอย่าง Sequence Diagram**



**ตัวอย่าง**

จงเขียน Sequence Diagram ของ Use Case พิมพ์งาน ซึ่งจะมีการส่งเอกสารที่ต้องการพิมพ์จาก Computer ไปยัง Print Server ซึ่งจะตรวจสอบว่า Printer ว่างหรือไม่ถ้าขณะนั้น Printer ว่างก็จะส่งเอกสารไปยัง Printer เพื่อพิมพ์ในทันที แต่ถ้า Printer ไม่ว่างก็จะส่งไปเก็บใน Queue ก่อนเพื่อรอให้ Printer ว่าง



**Collaboration Diagram**

Collaboration Diagramเป็นแผนภาพซึ่งแสดงปฏิสัมพันธ์(Interaction) ระหว่าง Class หรือ Object ตามลำดับขั้นตอนของเหตุการณ์ จะเน้นอธิบายโครงสร้างของ Object ว่ามีการรับส่ง Massage กันอย่างไร ซึ่งเป็นการแสดงความร่วมมือของ Class หรือ Object อย่างชัดเจน

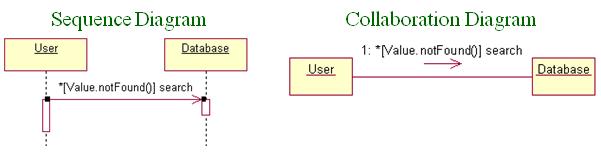
มีหน้าที่เดียวกับ Sequence Diagram แต่จะไม่แสดงถึงแกนเวลาอย่างชัดเจน

**ขั้นตอนการสร้าง Collaboration Diagram**

* + พิจารณาทีละ Use Case
  + ลำดับเหตุการณ์ก่อน-หลัง
  + พิจารณาเหตุการณ์ว่ามีความเกี่ยวข้องสัมพันธ์ของ Class/Object ใดก่อนหลัง
  + เขียน Actor ที่ทำให้เกิด Use Case นั้น
  + เขียน Class/Object ที่เกิดใน Use Case
  + ลากเส้นเชื่อม (Link) ความสัมพันธ์ระหว่าง Class และ ใช้เส้นลูกศรกำกับ พร้อมระบุลำดับเหตุการณ์ด้วยตัวเลข กรณีมีเหตุการณ์ย่อยใช้ค่าทศนิยม

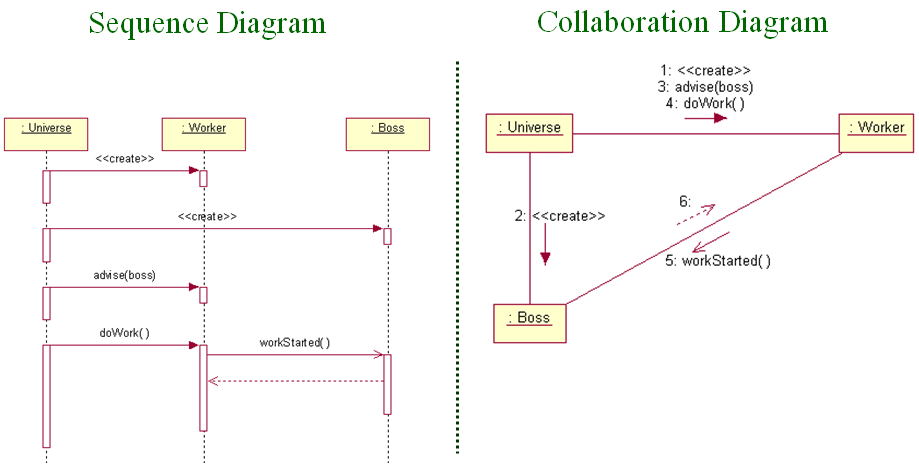
**เพิ่มเติม Control Information**

* ในส่วนแรกที่กล่าวถึง การระบุรายละเอียดของ Message
* ในรูปแบบ [Guard Condition]Message Name
* สามารถระบุการทำซ้ำโดยใช้รูปแบบ
* \* [Guard Condition] Message Name
* คือ Message ถูกส่งออกไปหลายๆครั้ง
* และ Object ที่รับอาจมีหลายๆ Object

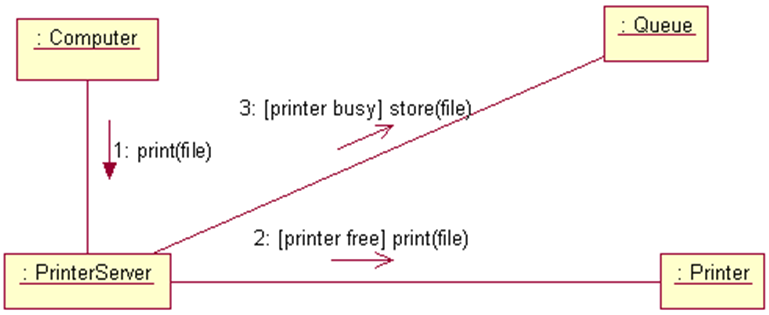


**Interaction Modeling**

**ตัวอย่าง**



**ตัวอย่าง Collaboration Diagram**



**ข้อแตกต่างระหว่าง** Sequence diagram และ Collaboration diagram

Sequence diagram แสดงรายละเอียดแต่ละ Use Case ว่า Object แต่ละตัวติดต่อสื่อสารกันอย่างไร และมีลำดับขันตอนการทำงานอย่างไร โดยเน้นแกนเวลาเป็นสำคัญ

Collaboration diagram มีหน้าที่เช่นเดียวกับ Sequence diagram คือใช้แสดงปฏิสัมพันธ์ระหว่าง Object ตามลำดับขั้นตอนของเหตุการณ์ แต่จะแตกต่างจาก Sequence diagram ตรงที่ จะไม่แสดงแกนเวลาของ Object แต่จะเน้นการอธิบายโครงสร้างของ Object ว่ามีการรับส่ง Message ระหว่าง Object อย่างไร ซึ่งจะทำให้เห็นภาพรวมว่ามีเหตุการณ์ใดบ้างเกิดขึ้นกับแต่ละ Object และจะมีตัวเลขกำกับ Message เพื่อแสดงให้เห็นลำดับของเหตุการณ์

Collaboration diagram มีหน้าที่เช่นเดียวกับ Sequence diagram คือใช้แสดงปฏิสัมพันธ์ระหว่าง Object ตามลำดับขั้นตอนของเหตุการณ์ แต่จะแตกต่างจาก Sequence diagram ตรงที่ จะไม่แสดงแกนเวลาของ Object แต่จะเน้นการอธิบายโครงสร้างของ Object ว่ามีการรับส่ง Message ระหว่าง Object อย่างไร ซึ่งจะทำให้เห็นภาพรวมว่ามีเหตุการณ์ใดบ้างเกิดขึ้นกับแต่ละ Object และจะมีตัวเลขกำกับ Message เพื่อแสดงให้เห็นลำดับของเหตุการณ์

**การเลือกใช้ Interaction Diagram**

* + ใช้ Sequence diagram เพื่อแสดงลำดับระหว่างสิ่งที่มากระตุ้นให้เกิดปฏิสัมพันธ์ ระหว่างวัตถุ มักใช้ใน Real-time modeling
  + ใช้ Collaboration diagram เมื่อโครงสร้างของระบบ มีความสำคัญ

**แบบฝึกหัด**

จงเขียน Collaboration Diagramของ Use Case พิมพ์งานซึ่งจะมีการส่งเอกสารที่ต้องการพิมพ์จาก Computer ไปยัง Print Server ซึ่งจะตรวจสอบว่า Printerว่างหรือไม่ถ้าขณะนั้น Printerว่างก็จะส่งเอกสารไปยัง Printer เพื่อพิมพ์ในทันที แต่ถ้า Printer ไม่ว่างก็จะส่งไปเก็บใน Queue ก่อนเพื่อรอให้ Printerว่าง