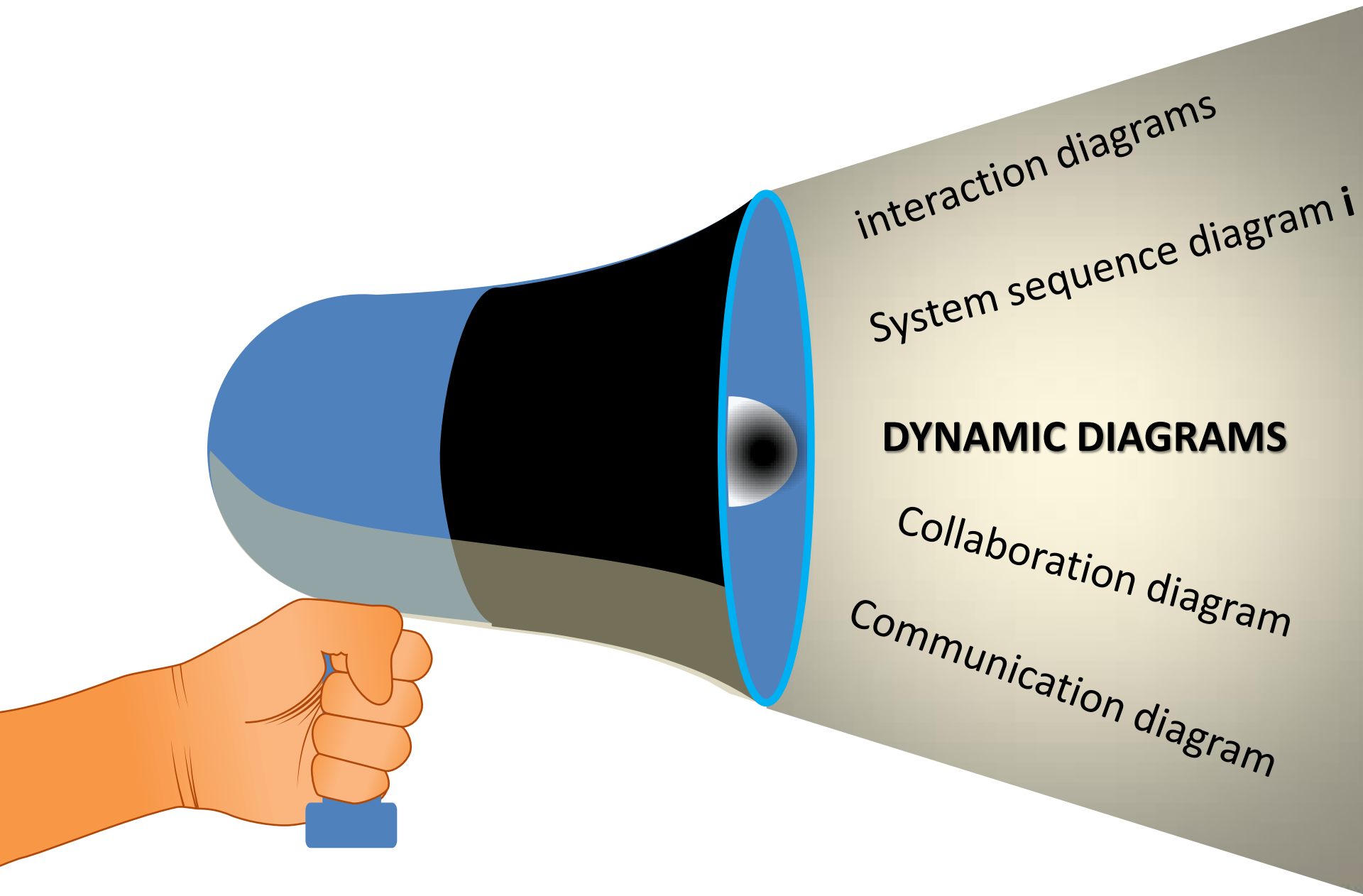
Module -5 Dynamic Diagrams

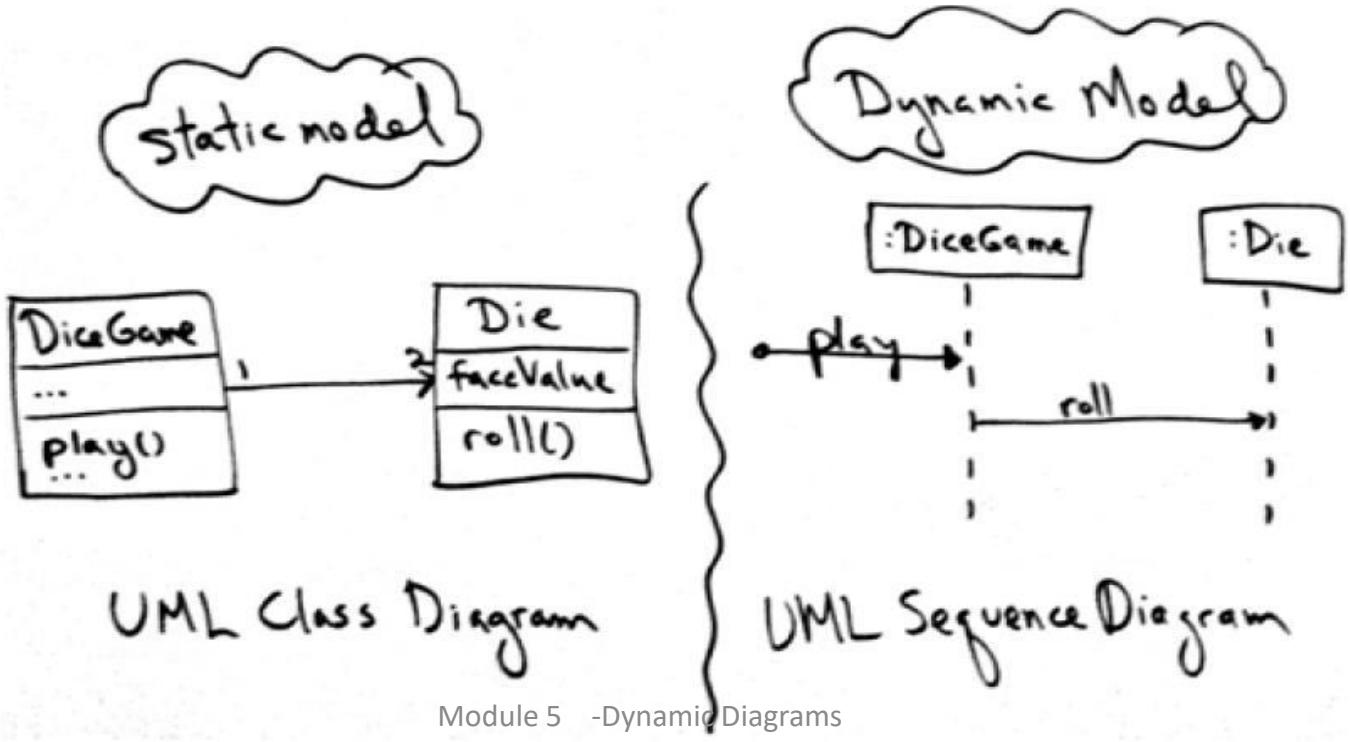
- interaction diagrams
-System sequence diagram
–Collaboration diagram
-Communication diagram

Dr.S.Neelavathy Pari
Assistant Professor (Sr. Grade)
Department of Computer Technology
Anna University, MIT Campus


interaction diagrams
System sequence diagram i
DYNAMIC DIAGRAMS
Collaboration diagram
Communication diagram

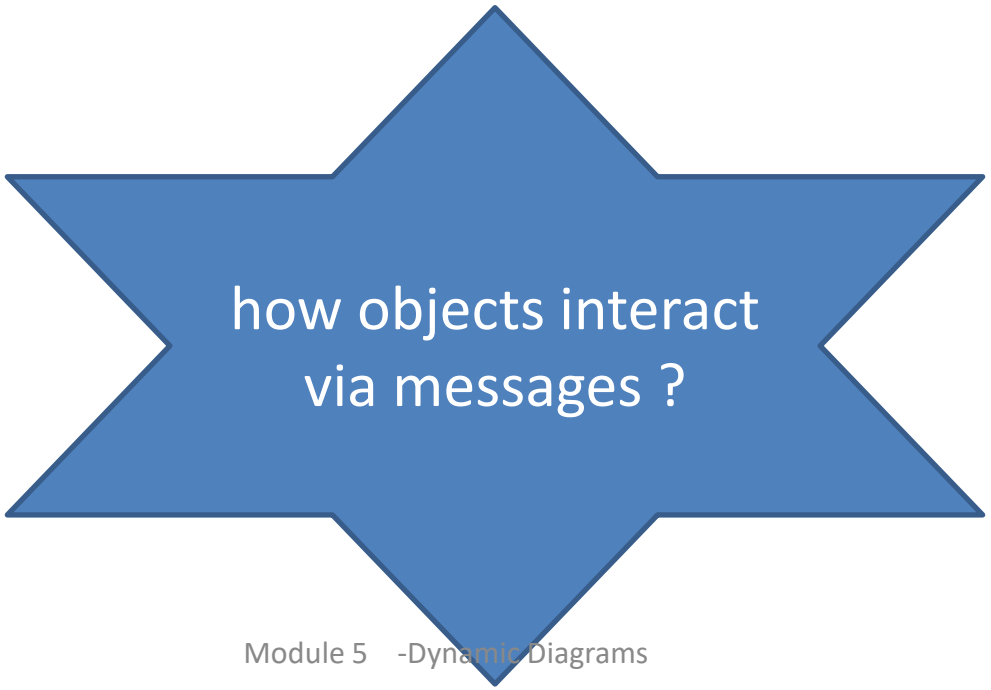
Dynamic Diagrams

- Design the logic,
- Behavior of the code
- Method bodies.



UML Interaction Diagrams

- Provide a reference for frequently used UML interaction diagram notation sequence and communication diagrams.

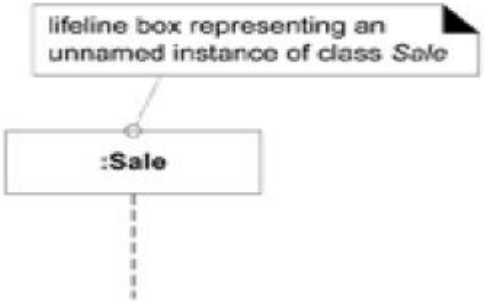


interaction diagram

- The term **interaction diagram is a generalization of two more specialized UML diagram types:**
- sequence diagrams
- communication diagrams

Different Types of Interaction Diagrams

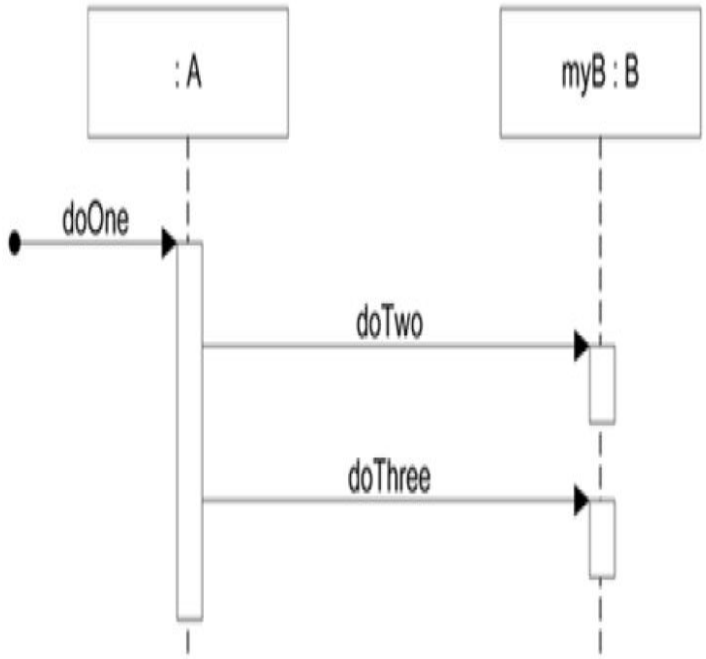
- An Interaction Diagram typically captures a use-case
  - A sequence of user interactions
- **Sequence diagrams**
  - Highlight the sequencing of the intera objects
- Collaboration diagrams
  - Highlight the structure of the components (objects) involved in the interaction



Sequence Diagram

class *A has a method named doOne and an attribute of* type *B. Also, that class B has methods named doTwo and doThree*

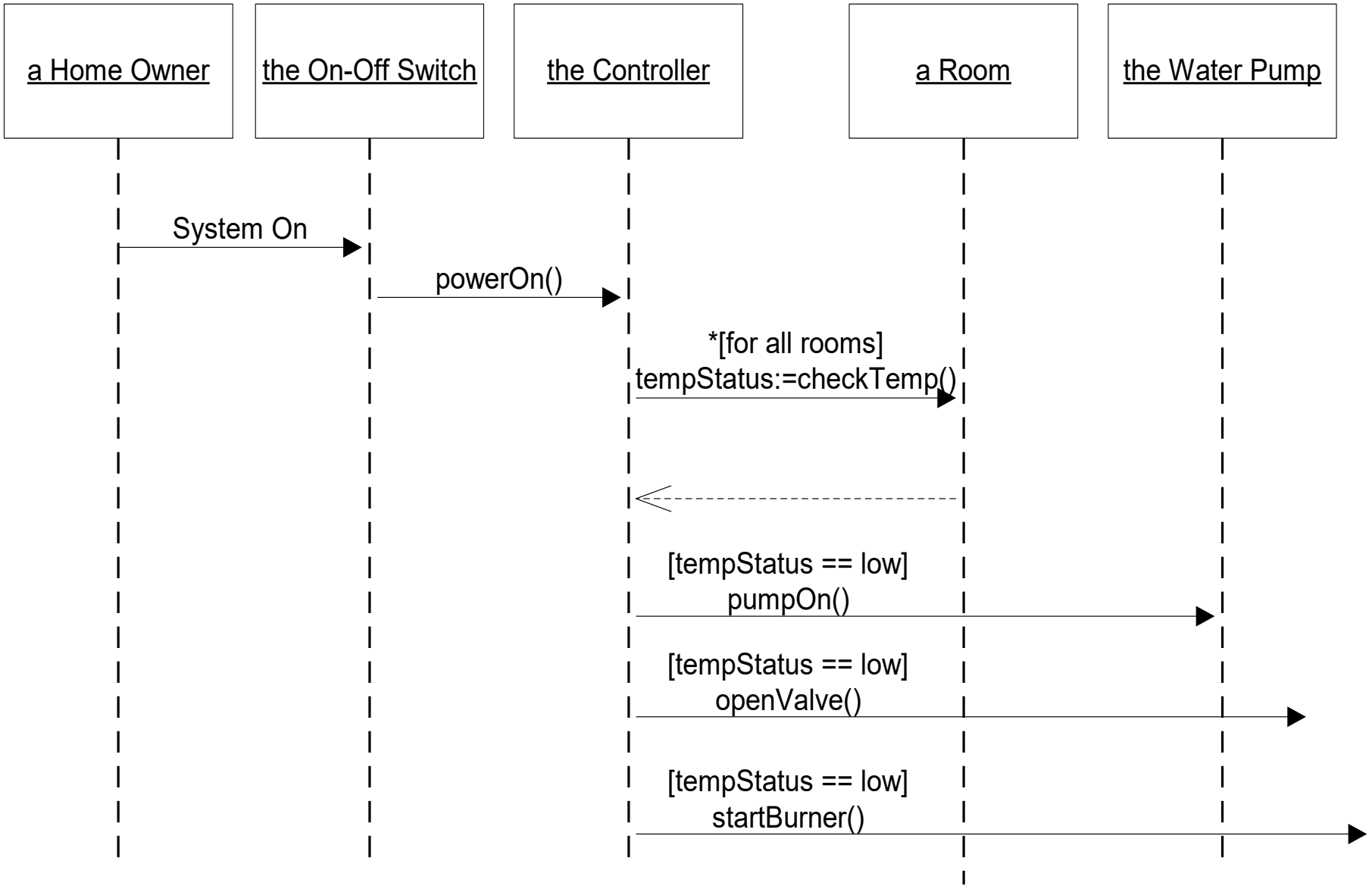
```
public class A
{
private B myB = new B();

public void doOne()
{
   myB.doTwo();
   myB.doThree();
}
// …
```



Home Heating Use-Case

Use case: **Power Up**
Actors: Home Owner (initiator)
Type: Primary and essential
Description: The Home Owner turns the power on. Each room is temperature checked. If a room is below the the desired temperature the valve for the room is opened, the water pump started, the fuel valve opened, and the burner ignited.
If the temperature in all rooms is above the desired temperature, no actions are taken.
Cross Ref.: Requirements XX, YY, and ZZ
Use-Cases: None

Sequence Diagrams



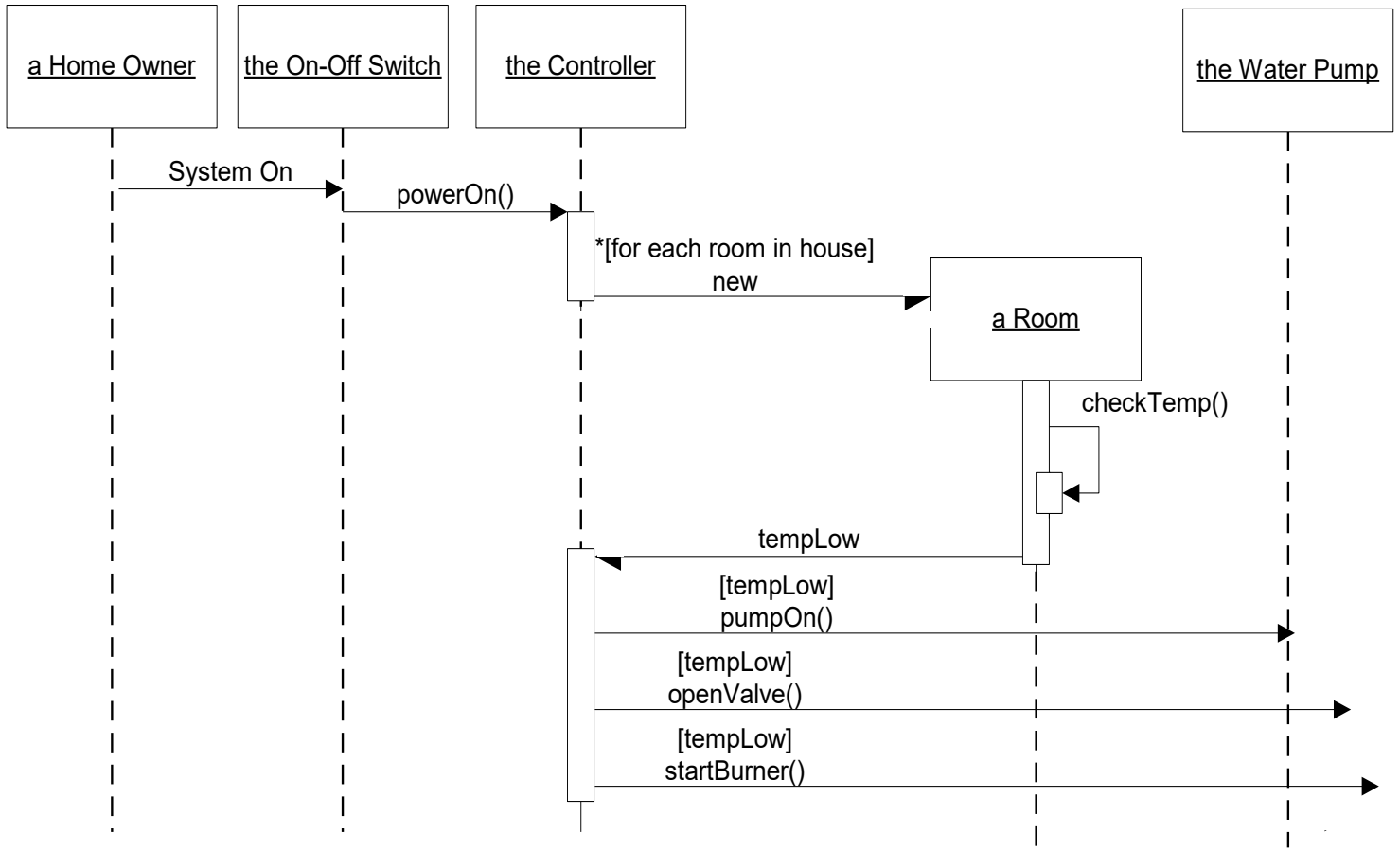
Comment the Diagram

When the owner turns the system on

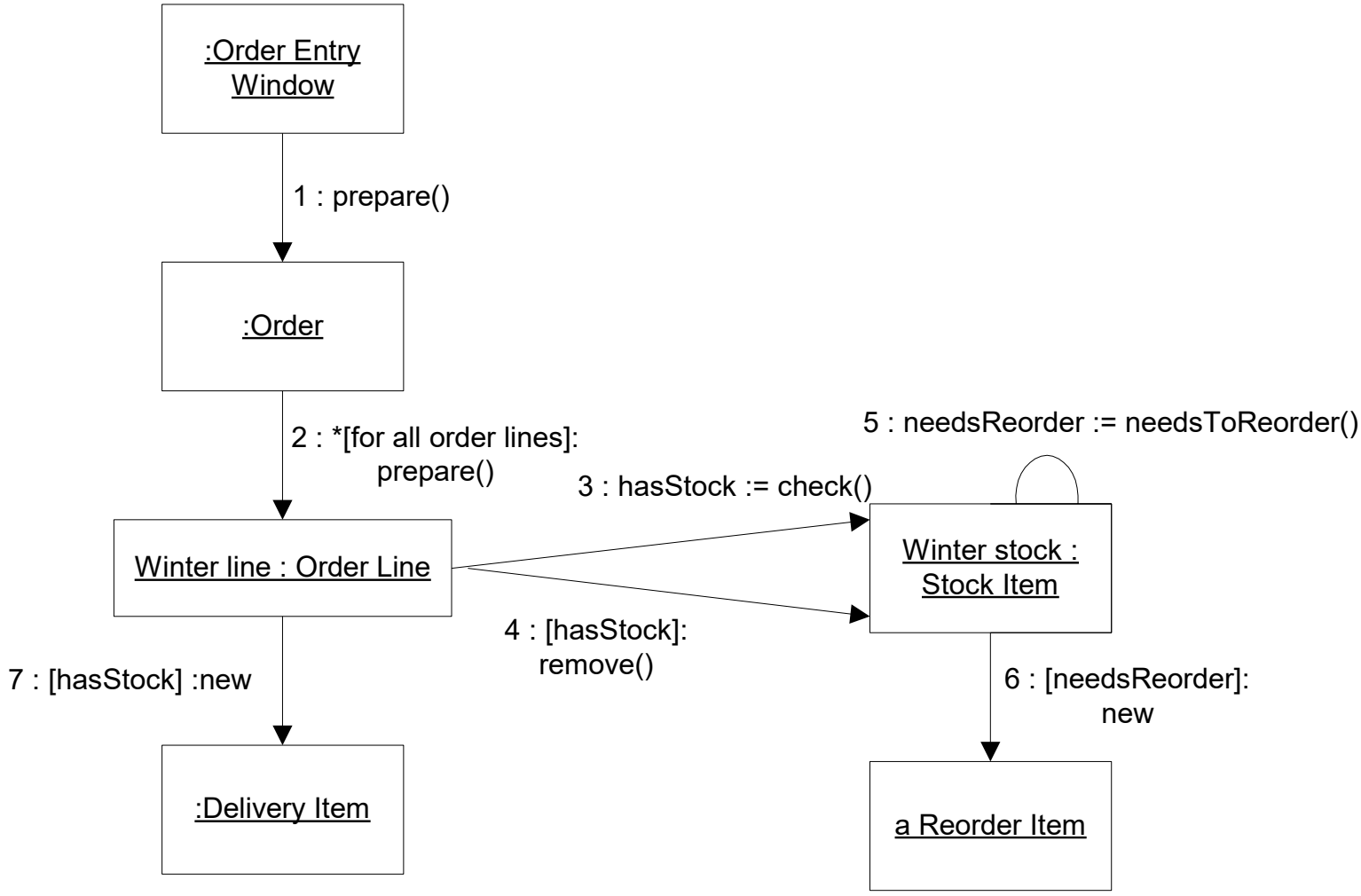
the on switch notifies the controller

The controller creates a room object for each room in the building

The rooms sample the temperature in the toom every 5 s. When a low temp is detected the room notifies the controller.



Collaboration Diagrams



Conditional Behavior

- Something you will encounter trying to capture complex use-cases
  - The user does something. If this something is X do this… If this something is Y do something else… If this something is Z…
- Split the diagram into several
  - Split the use-case also
- Use the conditional message
  - Could become messy
- ***Remember, clarity is the goal!***

Comparison

- Both diagrams capture the same information
  - People just have different preferences
- We prefer sequence diagrams
  - They clearly highlight the order of things
  - Invaluable when reasoning about multi-tasking
- Others like collaboration diagrams
  - Shows the static structure
    - Very useful when organizing classes into packages
- We get the structure from the Class Diagrams

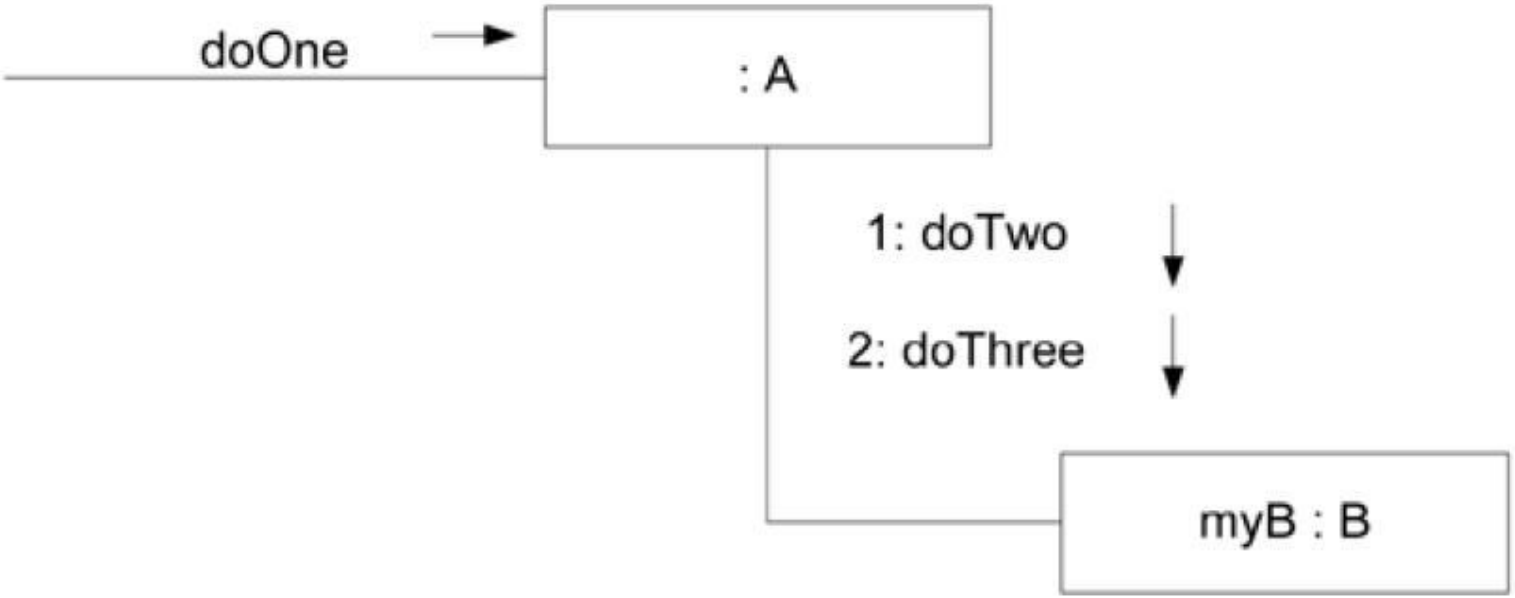
When to Use Interaction Diagrams

- When you want to clarify and explore single use-cases involving several objects
  - Quickly becomes unruly if you do not watch it
- If you are interested in one object over many use-cases -- **state transition diagrams**
- If you are interested in many objects over many use cases -- **activity diagrams**

Communication diagrams

- **Communication diagrams illustrate object interactions in a graph or network format, in which objects can be** placed anywhere on the diagram (the essence of their wall sketching advantage),

Communication diagrams



Identify the Strength and weakness of Sequence and Communication diagram

some related code for the *Sale class and its makePayment method*

```
public class Sale
{
private Payment payment;

public void makePayment( Money cashTendered )
{
    payment = new Payment( cashTendered );
    //…
}
// …
}
```

Construct sequence and communication diagram