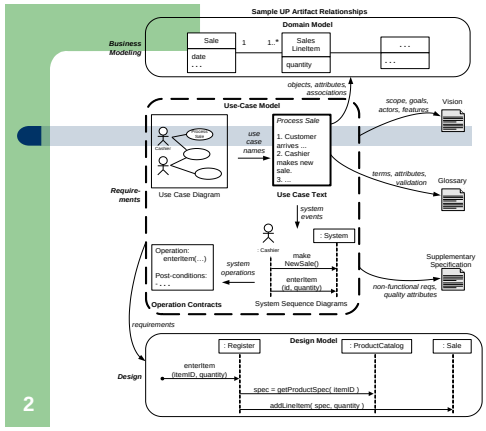
面向对象程序的分析与设计
Object-Oriented Analysis and Design

Lecture 7
OOD (I)

Prof. S. Xu



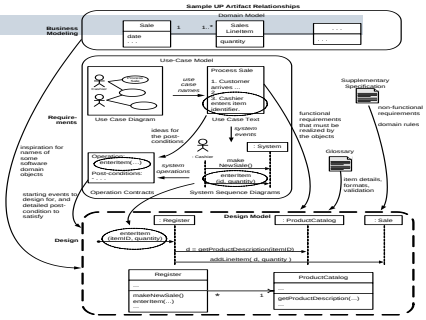
Contents

- OOD
- Interaction Diagrams

OOD (Object Oriented Design)

On to Object Design

- We are on our way to designing collaborating objects to fulfill project requirement.

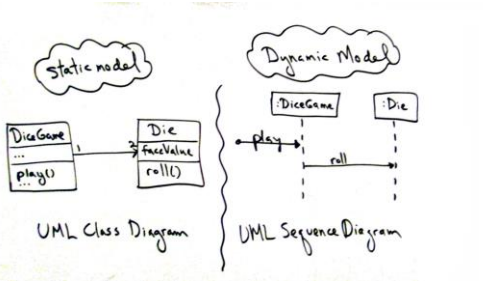


Designing Objects: Static and Dynamic Modeling

- There are two kinds of object (Design) models: ***dynamic and static***.
- Static model includes Design Class Diagrams and package diagrams
- Dynamic model includes UML interaction diagrams (sequence diagrams or communication/collaboration diagrams)

Designing Objects: Static and Dynamic Modeling

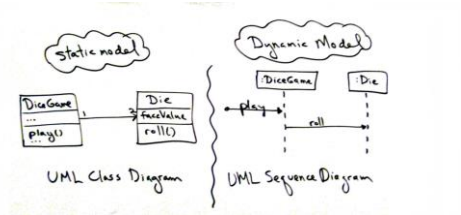
- ***Static models***, such as *UML class diagrams*, help design the definition of packages, class names, attributes, and method signatures (but not method bodies).



7

Definition: Design Class Diagram

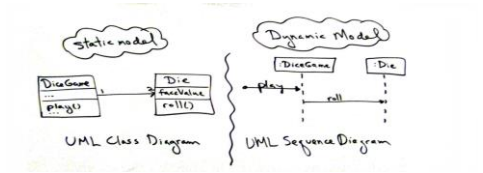
- We need a unique term to clarify when the class diagram is used in a software or design perspective.
  - A common modeling term for this purpose is ***design class diagram (DCD)***.



8

Designing Objects: Static and Dynamic Modeling

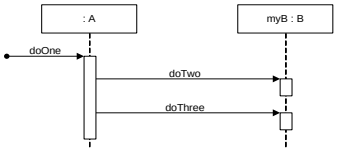
- During dynamic modeling, we apply ***responsibility-driven design*** and the ***GRASP principles.***
- Important to stress:
  - ***What's important is knowing how to think and design in objects, and apply object design best-practice patterns,***



9

Designing Objects: Static and Dynamic Modeling

- While drawing a UML object diagram, we need to answer key questions:
  - ***What are the responsibilities of the object?***
  - ***Who does it collaborate with?***
  - ***What design patterns should be applied?***



10

Review

- When arranging actors and objects on a sequence diagram, it is nice to list them
- _____.
  - a. in alphabetical order down the side of the diagram
  - b. in alphabetical order across the top of the diagram
  - c. in order in which they participate in the sequence down the side of the diagram
  - d. in order in which they participate in the sequence across the top of the diagram

11

Review

- Which of the following is false about Use Case Analysis?
  - A: An Actor must be directly associated with at least one use case in the system.
  - B: A use case must be directly related to at least one actor of the system
  - C: A use case can be related to actors and other use cases.
  - D: Actors can be people, machines, or other systems

12

Review

- Information used to develop use case diagrams comes from:
  - A. Class Diagrams
  - B. Sequence Diagrams
  - C. Designers best guess
  - D. Customer Requirements

13

Review

- What diagram(s) are included in dynamic Design modelling?
- What diagram(s) are included in static Design modelling?

14

Review

- In UML Class diagrams, there are three relationships between classes: generalization, association and dependency. Explain the differences and give examples (showing them with UML diagram)

15

Review

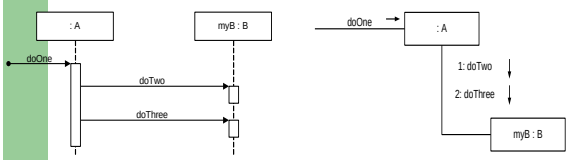
- Health Future Vision

Interaction Diagrams

17

UML Interaction Diagrams

- ***Interaction diagrams*** are to illustrate how objects interact via messages.
- There are two common types:
  - *sequence diagrams.*
  - *communication diagrams (collaboration diagrams).*



18

UML Interaction Diagrams

- Sequence diagrams illustrate interactions in a kind of fence format, in which each new object is added to the right:

What might this represent in code?

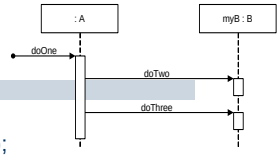
Probably, that class A has a method named doOne and an attribute of type B.

Also, that class B has methods named doTwo and doThree.

Perhaps the partial definition of class A is:

19

UML Interaction Diagrams

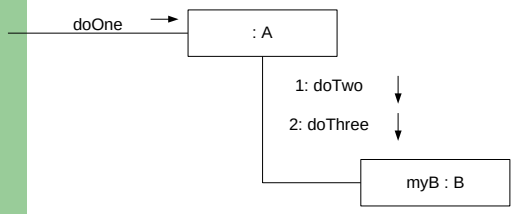



```
blic class A
{
   private B myB = new B();
   public void doOne()
   {
      myB.doTwo();
      myB.doThree();
   } // …
}
```

20

UML Interaction Diagrams

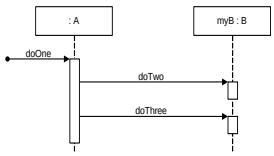
- ***Communication diagrams*** illustrate object interactions in a graph or network format, in which objects can be placed anywhere on the diagram.



21

Sequence v. Communication Diagrams

- Sequence diagrams have some advantages over communication diagrams.
  - It is easier to see the call-flow sequence with sequence diagrams - simply read top to bottom.
  - With communication diagrams we must read the sequence numbers, such as "1:" and "2:".
  - Hence, sequence diagrams are excellent for documentation or to easily read a reverse-engineered call-flow sequence, generated from source code with a UML tool.



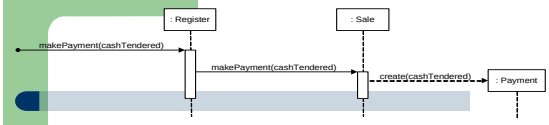
22

Sequence v. Communication Diagrams

Type	Strengths	Weaknesses
sequence	clearly shows sequence or time ordering of messages large set of detailed notation options	forced to extend to the right when adding new objects; consumes horizontal space
communication	space economicalflexibility to add new objects in two dimensions	more difficult to see sequence of messages fewer notation options

23

Example Sequence Diagrams

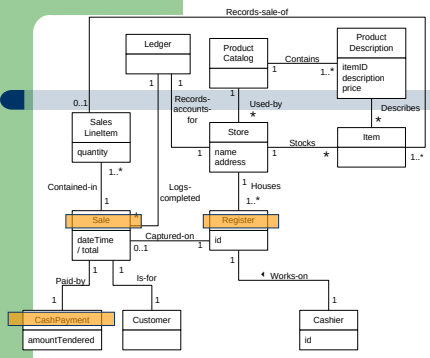



```
public class Sale {
        private Payment payment;

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

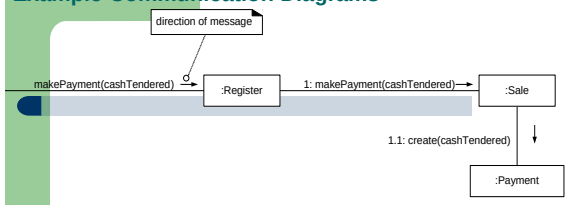
24

Example Sequence Diagrams



25

Example Communication Diagrams



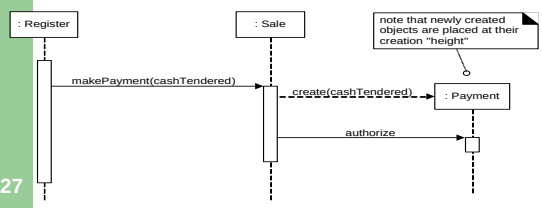
- **Spend time doing dynamic object modeling with interaction diagrams, not just static object modeling with class diagrams**

26

Basic Message Expression Syntax – Sequence Diagram

- Interaction diagrams show messages between objects; the UML has a standard syntax for these message expressions:

return = message(parameter : parameterType) : returnType



27

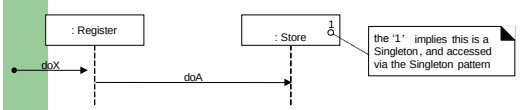
Basic Message Expression Syntax

- For example:
  - d = getProductDescription(id)
  - d = getProductDescription(id:ItemID)
  - d = getProductDescription(id:ItemID) : ProductDescription

28

Singleton Object

- Sometimes, there is only one instance of a class instantiated, never two.
  - it is a ***"singleton"*** instance.
- "Singleton" object is marked with a '1' in the upper right corner of the lifeline box



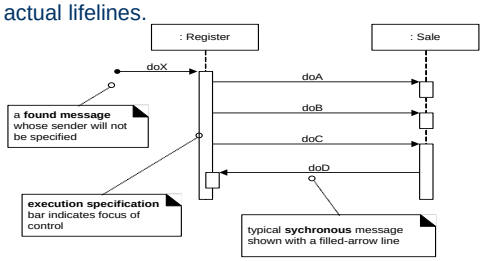
All these notations apply to both kinds of interaction diagrams.

29

Basic Sequence Diagram Notation

Lifeline Boxes and Lifelines

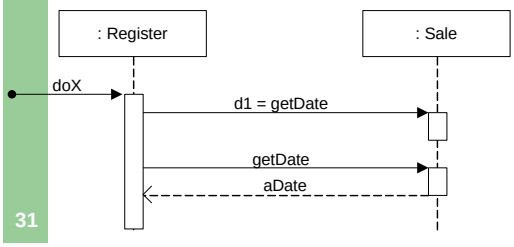
- In contrast to communication diagrams, in sequence diagrams the lifeline boxes include a vertical line extending below them- these are the actual lifelines.



30

Illustrating Reply or Returns

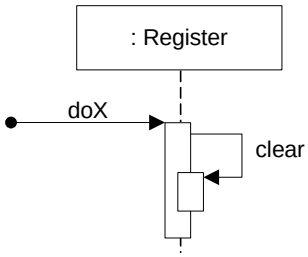
- There are two ways to show the return result from a message:
  - Using the message syntax returnVar = message(parameter).
  - Using a reply (or return) message line at the end of an execution spec. bar.



31

Message to "self" or "this"

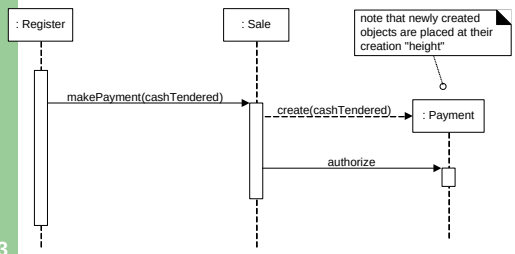
- You can show a message being sent from an object to itself by using a nested Exec. Spec. bar



32

Creation of Instance

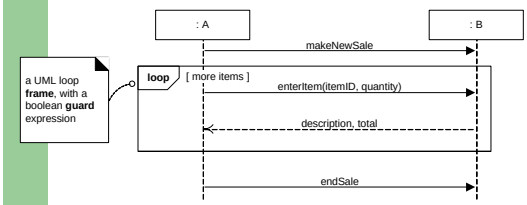
- Using a ***dashed line*** with a filled arrow to represent "call the constructor".



33

Diagram Frames in UML Sequence Diagram

- **To support conditional and looping constructs (among many other things), the UML uses frames.**



34

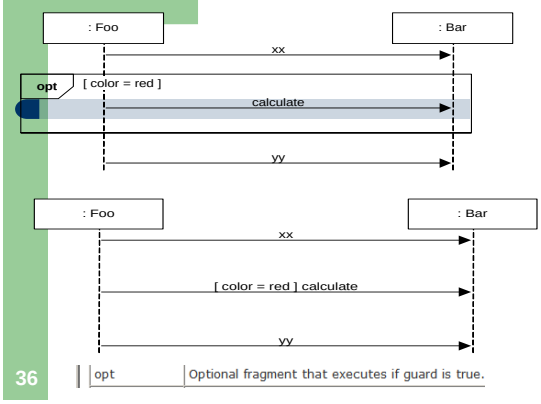
Diagram Frames in UML Sequence Diagram

The following table summarizes some common frame operators:

Frame Operator	Meaning
alt	Alternative fragment for mutual exclusion conditional logic expressed in the guards.
loop	Loop fragment while guard is true. Can also write *loop(n)* to indicate looping n times. There is discussion that the specification will be enhanced to define a *FOR* loop, such as *loop(i, 1, 10)*
opt	Optional fragment that executes if guard is true.
par	Parallel fragments that execute in parallel.
region	Critical region within which only one thread can run.

35

Diagram Frames in UML Sequence Diagram

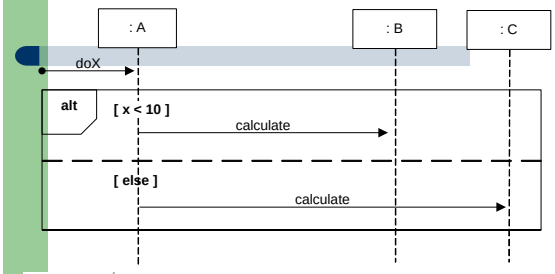


36

opt | Optional fragment that executes if guard is true.

Diagram Frames in UML Sequence Diagram

Basic Sequence Diagram Notation

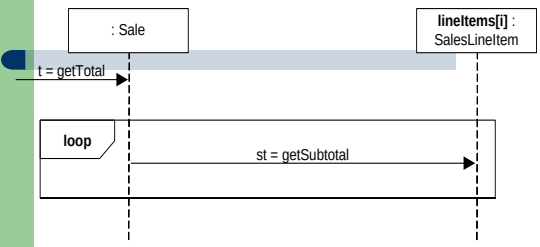


alt — Alternative fragment for mutual exclusion conditional logic expressed in the guards.

37

Diagram Frames in UML Sequence Diagram

Basic Sequence Diagram Notation

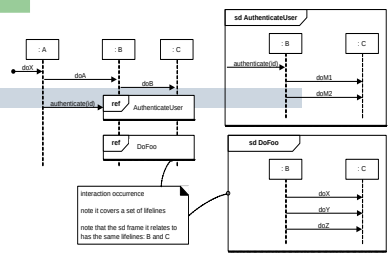


loop — Loop fragment while guard is true. Can also write *loop(n)* to indicate looping n times. There is discussion that the specification will be enhanced to define a *FOR* loop, such as *loop(i, 1, 10)*

38

How to Relate Interaction Diagrams

Basic Sequence Diagram Notation

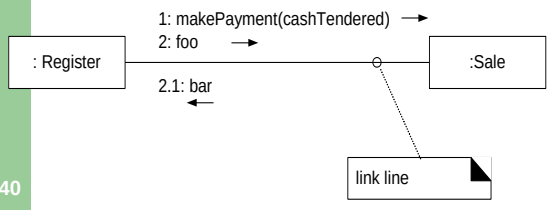


- They are created with two related frames:
  - a frame around an entire sequence diagram , labeled with the tag ***sd*** and a name, such as AuthenticateUser.
  - a frame tagged ***ref***, called a reference, that refers to another named sequence diagram; it is the actual interaction occurrence.

39

Basic Communication Diagram Notation - Link

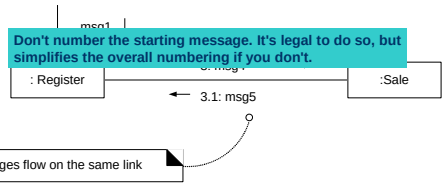
- A **link** is a connection path between two objects; *a link is an instance of an association.*
  - For example, there is a link - from a Register to a Sale,.



40

Message

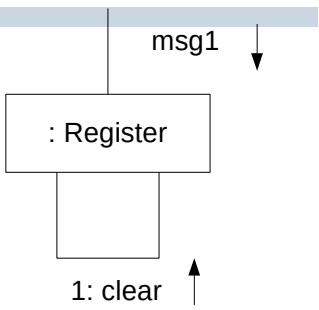
- Each ***message*** between objects is represented with a message expression and small arrow indicating the direction of the message.
- A sequence number is added to show the sequential order of messages



41

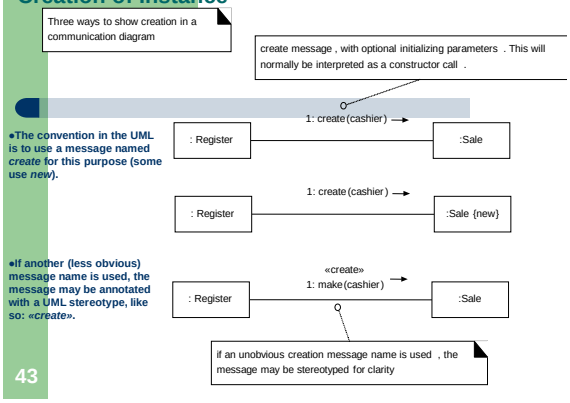
Message to "self" or "this"

- A message can be sent from an object to itself.



42

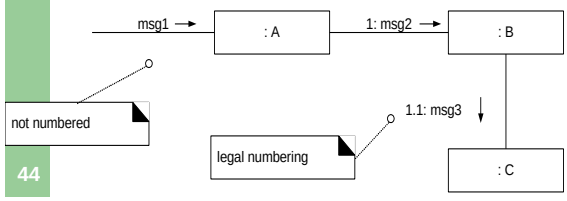
Creation of Instance



43

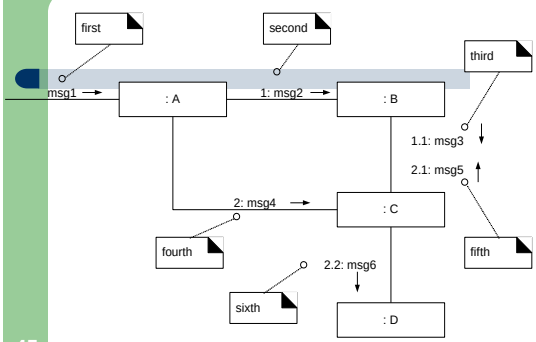
Message Numbering

- The order of messages is illustrated with sequence numbers. The numbering scheme is:
  - The order and nesting of subsequent messages is shown with a legal numbering scheme in which nested messages have a number appended to them.



44

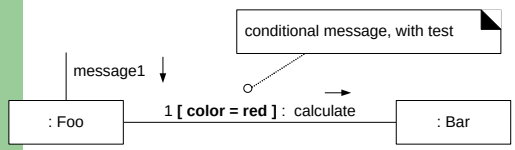
Message Numbering – A More Complex Case



45

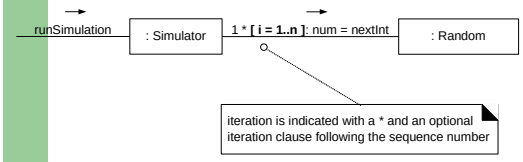
Conditional Message

- You show a conditional message by following a sequence number with a conditional clause in square brackets, similar to an iteration clause.



46

Iteration or Looping



47

Review

- City Future Vision

- Microsoft Future

49

Review

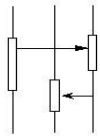
- Jacobson originally called these diagrams Object Interaction Diagrams. The notation has changed slightly in UML. What is the current UML term for these diagrams?

A. trace diagram

B. event trace diagram

C. sequence diagram

D. none of the above



50

Review

- How many of the following are UML diagram names ?
  A. collaboration diagram
  B. component diagram
  C. deployment diagram
  D. all of the above

51

Review

- When to use Interaction Diagrams?
  1. When you want to look at the behavior of several objects within a single use case or several use cases
  2. They are good at showing collaborations among objects; they are not so good at conditions and looping
  3. They are good to look at behavior of a single object across many use cases.
  4. All of the above
  5. None of the ABOVE

52

Review

- A software designer wants to show a scenario that has a time critical flow of control between multiple objects. What diagramming technique should they use?

A: Sequence Diagram
B: Collaboration Diagram
C: State Diagram
D: Class Diagram

53

Review

- A collaboration consists of _____.

a. two instances of a class talking with each other

b. two instances of a class knowing the value of each others attributes

c. a set of classes that share common operations

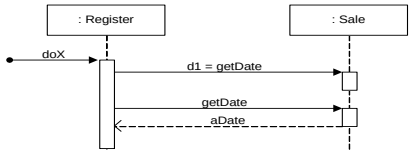
d. a set of classes that are all related to one another

54

Review

- What are the strengths and weakness of Interaction Diagrams? (Pick 2)

A. when you want to look at the behavior of several objects within a single use case
B. they are good at precise definition of the behavior
C. they are good at showing collaborations among objects
D. they are good at exploring concurrency and multi-thread issues



55

Review

- Which one of the following highlights the roles each object plays in an interaction model ? (Pick one).

A. Sequence Diagrams
B. Collaboration Diagrams
C. All of the above
D. None of the above

56

Review

- For showing how several objects collaborate in single use case, which one of the following OOAD artifacts is the MOST useful ? (Pick one).

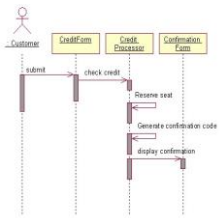
A. Interaction Diagrams
B. Activity Diagrams
C. Package Diagrams
D. State Diagrams
E. Class Diagrams

57

Review

- Refer to the diagram to answer the question. (Pick one. [MakePayment]. What methods MUST be implemented by the CreditProcessor class in the payment sequence diagram ?

A. checkCredit, generateConfirmationCode, displayCofirmation
B. checkCredit, generateConfirmationCode
C. checkCredit, generateConfirmationCode, reserveS
D. checkCredit, reserveSeat, displayCofirmation
E. checkCredit, reserveSeat



58

Review

Consider this Test program

```
Public class Test
{
    Public static void main (String [] args)
    {
        String s = "Hello World";
        StringTokenizer tokenizer = new StringTokenizer (s);
        While (tokenizer.hasMoreTokens ())
        {
            System.out.println (tokennizer.nextToken())
        }
    }
}
```

Draw a sequence diagram that shows the method calls of the main method

Review

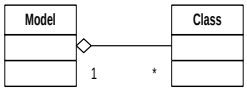
- **In UP, the author uses the term _____ _____ for a class diagram used in analysis. This is in contrast with a class diagram used in design, which he shockingly calls a _____ _____ _____.**

Review

- **Interaction diagrams are used for static/dynamic (circle one) object modeling.**
- **The two most common interaction diagrams are _____ diagrams and _____ diagrams.**

Review

A UML class diagram can be used to model UML itself by considering each graphical component of the notation to be a class. For example, a Model contains a collection of Class (as seen below).



Your problem is to construct and draw a class diagram with the following UML constructs: Model, Static Model, Dynamic Model, Class, Association, Generalization, Dependency, Aggregation, Composition, Multiplicity, Sequence Diagram, Message, and Collaboration Diagram.