Announcements, 3/23/2023

Today: UML Sequence Diagrams

Break around 11:15am

Acknowledgements

Some of these slides are based on the lecture notes from Prof. Alex Kuhn at SUNY Korea, Profs. Robert Kelly and Scott Stoller at Stony Brook University, and Martin Fowler's UML Distilled book.

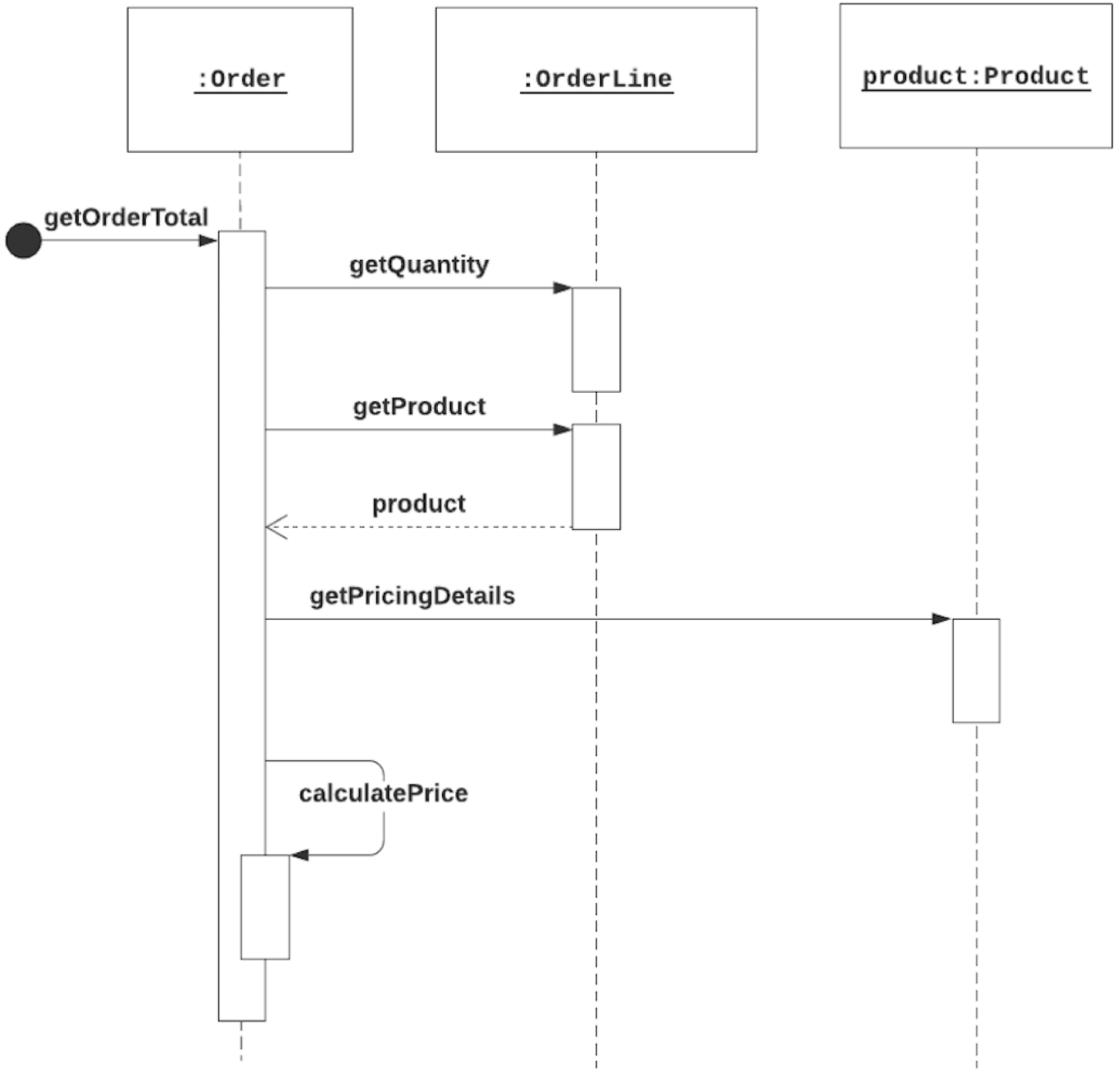
Outline

- What is a Sequence Diagram?
- Components of a Sequence Diagram
- Practice

What is Sequence Diagram?

What is a sequence diagram?

- Diagram that models a single scenario happening in the system
  - Relates directly to the use cases
- Shows interaction between different components
- Focuses on the sequences of messages sent between objects


:Order
:OrderLine
product:Product
getOrderTotal
getQuantity
getProduct
product
getPricingDetails
calculatePrice

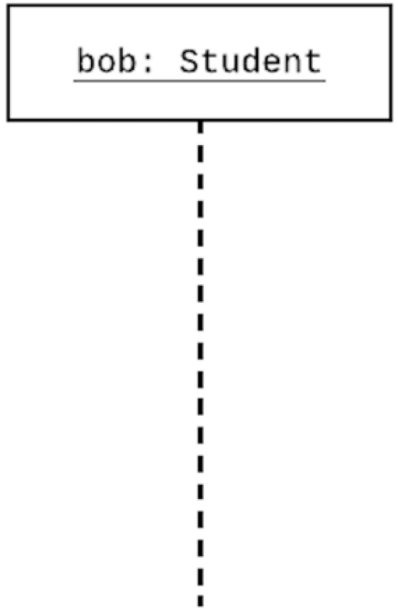
Why use sequence diagrams?

- To model how objects interact
- For software design:
  - Verify if we can support use cases with existing classes
  - Identify new classes and methods needed

Sequence Diagram Components

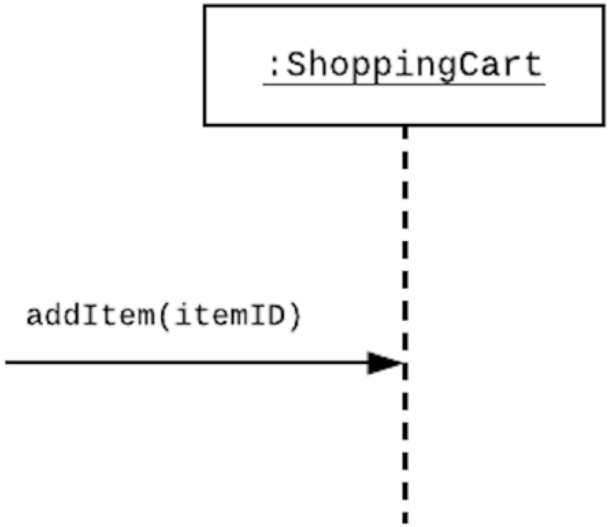
Representing an object

- Format is instance name : Class Name
- May only show either the class or instance name
  - Use instance name only if it clarifies the diagram
- A dotted line comes down from the top box to represent the object – this is called a **lifeline**



Messages

- Messages are when an object calls a method on another object
- Messages are shown as horizontal arrows
  - Label with method name and arguments
  - Arguments are optional, include if they are not obvious



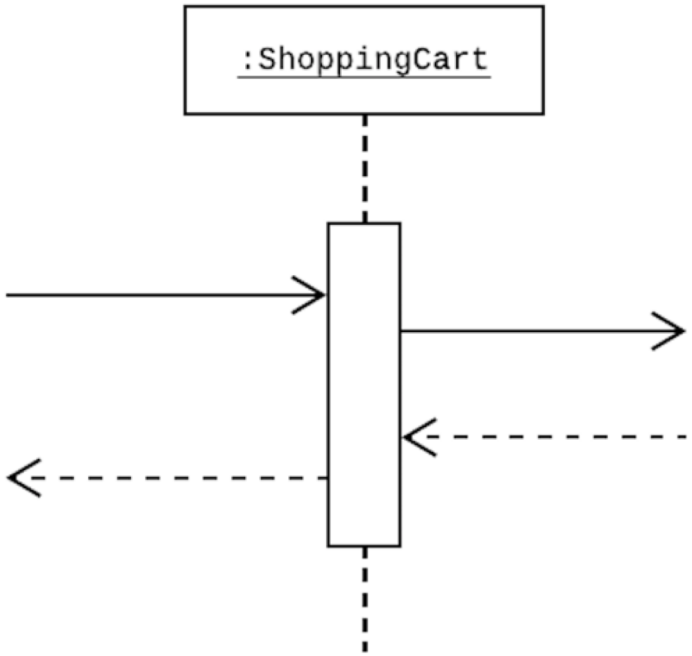
Types of messages

- Type of arrow indicates message
  - Solid arrowheads are synchronous messages
  - Open arrowheads are asynchronous messages
  - Return messages have dashed line with open arrowhead

Note: return messages can be omitted unless they add clarity to diagram

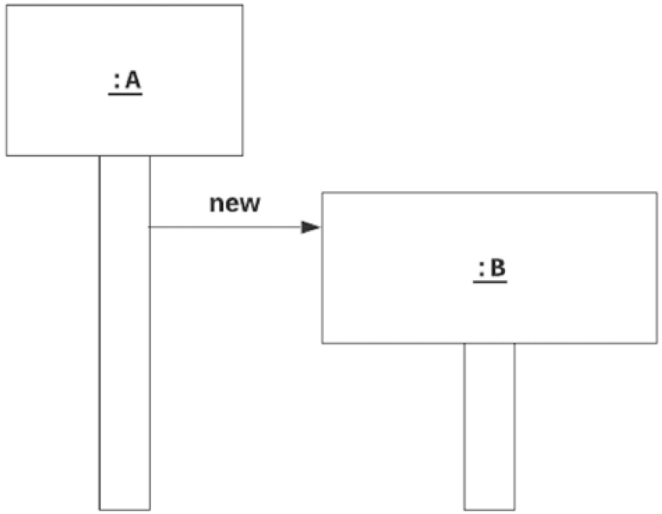
Activation

- There is a rectangular block when an object is activated
- Objects are activated when running code or on the stack waiting for a response
- Time passes as you move down the diagram



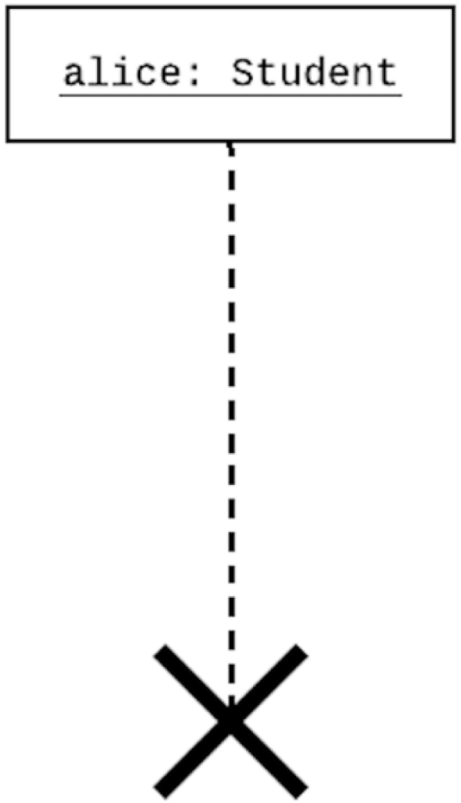
Object instantiation

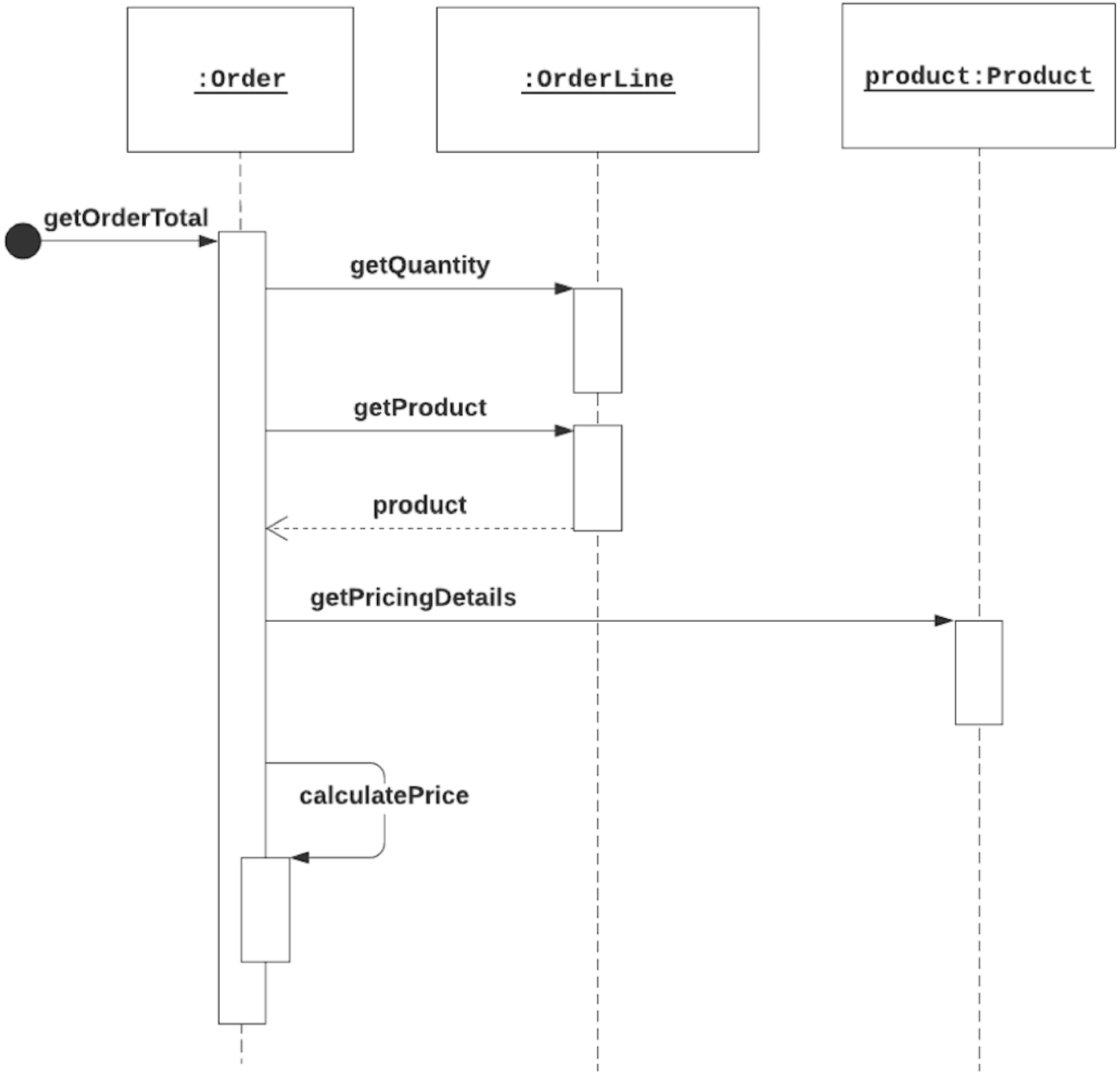
- An object can create another with a **new** message
- Sometimes an object is placed lower when it is created/activated during the use case



Deletion

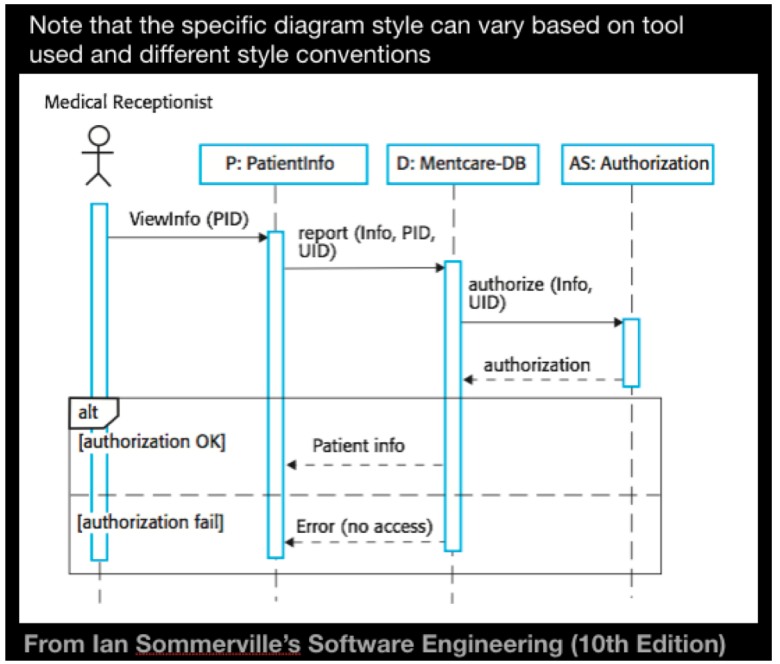
- If an object is deleted, can indicate with an X in the lifeline



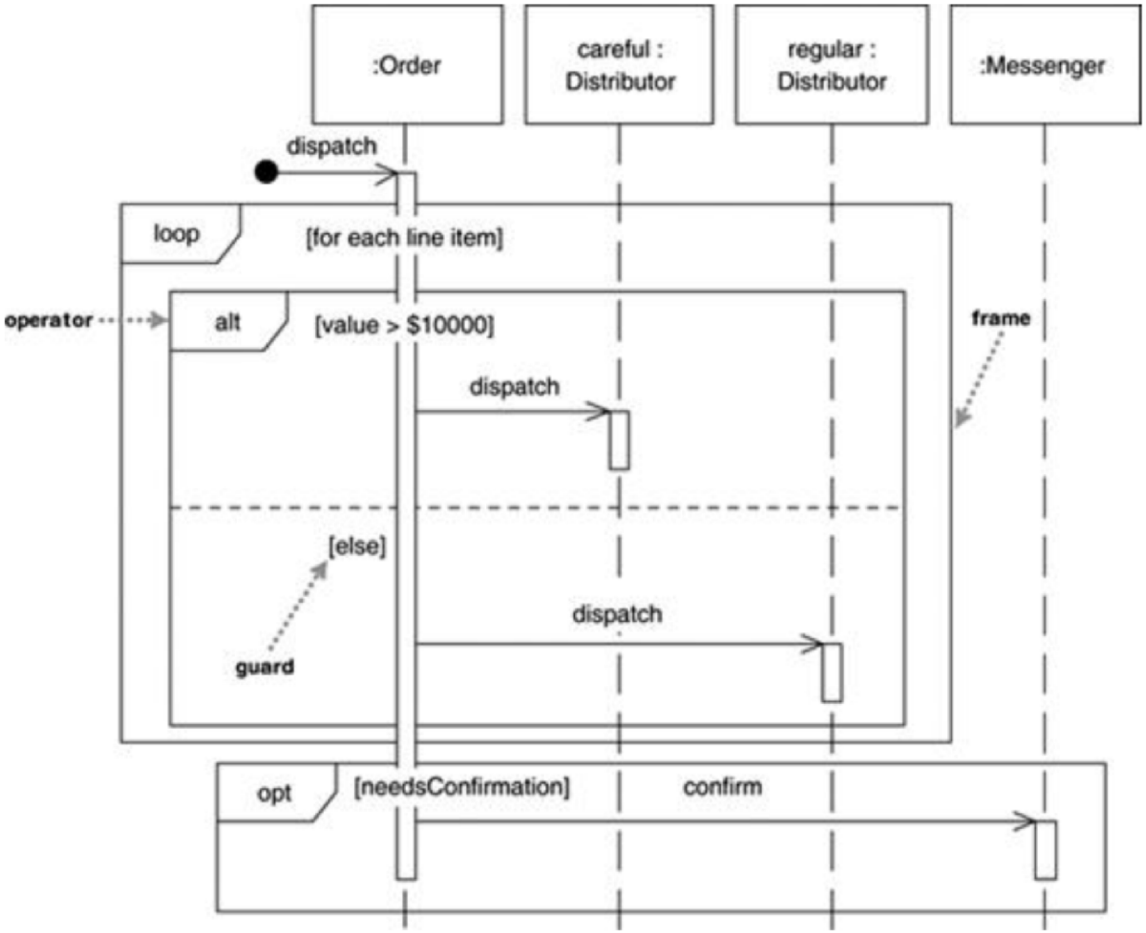

:Order
:OrderLine
product:Product
getOrderTotal
getQuantity
getProduct
product
getPricingDetails
calculatePrice

Loops and conditions

- Can use labeled frames to denote different conditions:
- if/else: **alt** [condition]
  – separated by horizontal dashed line
  –> see example
- if: **opt** [condition]
- loop: **loop** [condition or items to loop over]



Note: loops not very helpful to show – if needed, you could indicate with a comment

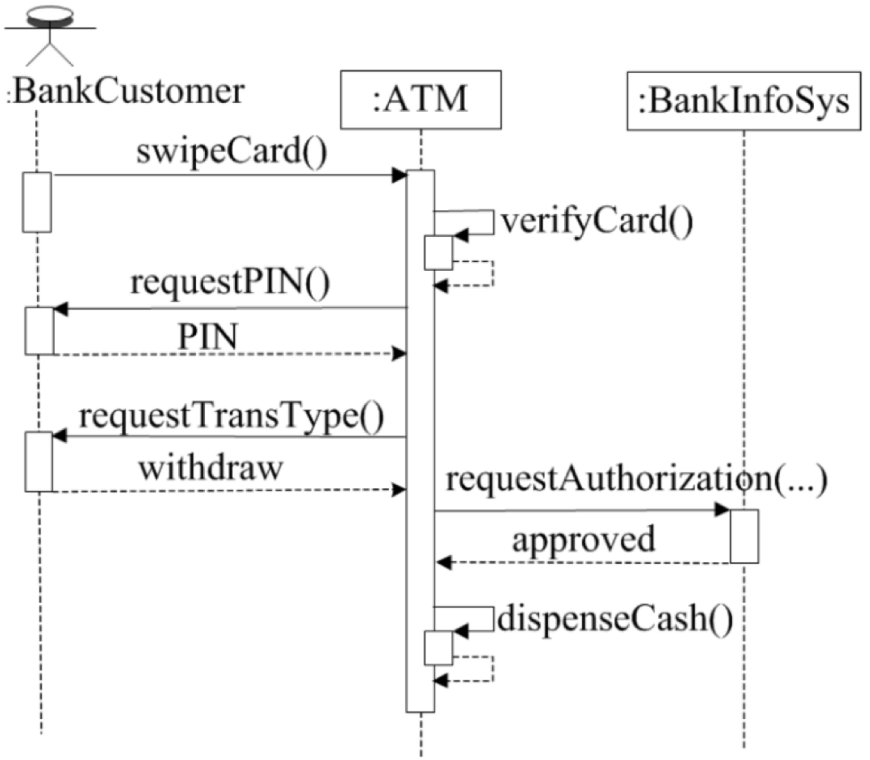


Pseudocode

```
procedure dispatch
  foreach (lineitem)
    if (product.value > $10K)
      careful.dispatch
    else
      regular.dispatch
    end if
  end for
  if (needsConfirmation)messenger.confirm
end procedure
```

Example sequence diagram for withdrawing cash

- Diagram shows interactions between the system (the ATM) and its environment.
- Could add lifelines for internal objects of ATM, e.g., transaction objects, if that level of detail is desired, but that goes beyond the use case.
- Nested rectangle is used to show call to verifyCard().
- This diagram shows only the main scenario of the use case. To show alternative flows, use alternative / optional construct or separate diagrams.



Design approach

- Work on the class diagram and sequence diagram concurrently
- Use your sequence diagrams to identify classes, attributes, and methods needed in your class diagram
- Modify your design as needed

General tips

- Creating first few sequence diagrams can be challenging
  - Requires understanding your design and framework philosophy
  - May need to review common patterns for the technologies you are using

Additional resources

- Sequence Diagram article: https://developer.ibm.com/articles/the-sequence-diagram/
- Sequence Diagram reference: https://www.lucidchart.com/pages/uml-sequence-diagram

Practice

Exercise: sequence diagrams for shopping

- Draw a sequence diagram for the primary flow of the "Shop" use case of the Internet Sales System (ISS) of the bookstore case study, as described on the next slide.
- Show customer actions and internal actions of the ISS.
- Include a Customer actor, a FrontEnd object, and internal objects of ISS that are needed (e.g., SearchController, ShoppingCart, OrderItem, Order, Authentication, ...)
  - For the internal objects, make up whatever objects that would be needed following the patterns for your chosen web technologies – state these technologies in a note on the diagram.
- The FrontEnd object reflects that the user does not interact directly with the ISS's internals.
- Don't show browser and web server explicitly. At this level of detail, Customer actor represents the customer and browser, and FrontEnd object represents the web server and front-end code.

Exercise: sequence diagrams for shopping

Shop use case: main scenario (simplified)

1. Customer submits search request to system.
2. System displays list of search results
3. Customer adds book to cart
4. Customer repeats steps 1-3 until finished shopping
5. Include Checkout use case

Checkout use case: main scenario (simplified)

1. System displays login page
2. Customer logs in
3. System displays shipping options page (mail order or in-store pickup)
4. Customer selects mail order and enters shipping address
5. System displays payment page
6. Customer enters payment information
7. System displays order information and asks customer to confirm order
8. Customer confirms order