CSC 2210
Object Oriented Analysis & Design

Dr. Akinul Islam Jony
Associate Professor
Department of Computer Science, FSIT
American International University - Bangladesh (AIUB)

akinul@aiub.edu

Sequence Diagram

>> What is Sequence Diagram?
>> Elements of Sequence Diagram
>> Extended Elements of Sequence Diagram
>> Why We Need Sequence Diagram?
>> Exercises

What is Sequence Diagram?

>> A **Sequence diagram** is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

>> A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

>> **Sequence** and **Collaboration** diagrams both illustrate the ***interactions*** *between objects.* Interactions show us how objects talk to each other.

>> Sequence Diagram:

- Illustrates how objects interacts with each other.
- **Emphasizes time ordering of messages.**
- Can model simple sequential flow, branching, iteration, recursion and concurrency.

>> Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

What is Sequence Diagram?

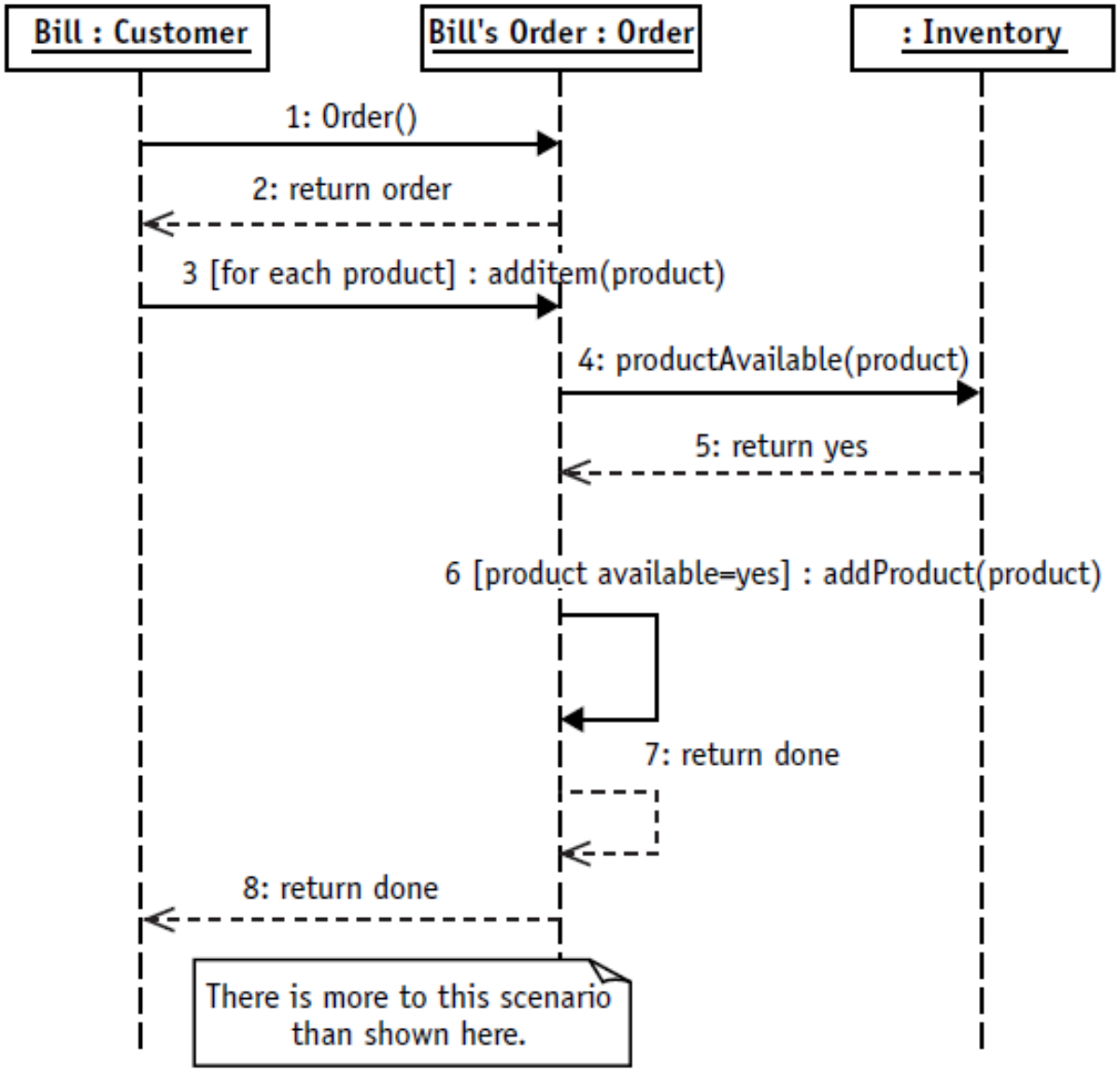


Figure 16-1 A basic Sequence diagram

What is Sequence Diagram?

>> Figure 16-1 shows a Sequence diagram with three participating objects:

- Bill (the Customer),
- Bill's Order, and
- the Inventory.

>> Without even knowing the notation formally, you can probably get a pretty good idea of what is going on.

- Steps 1 and 2: Bill creates an order.
- Step 3: Bill tries to add items to the order.
- Step 4 and 5: Each item is checked for availability in inventory.
- Step 6 and 7: If the product is available, it is added to the order.
- Step 8: He finds out that everything worked.

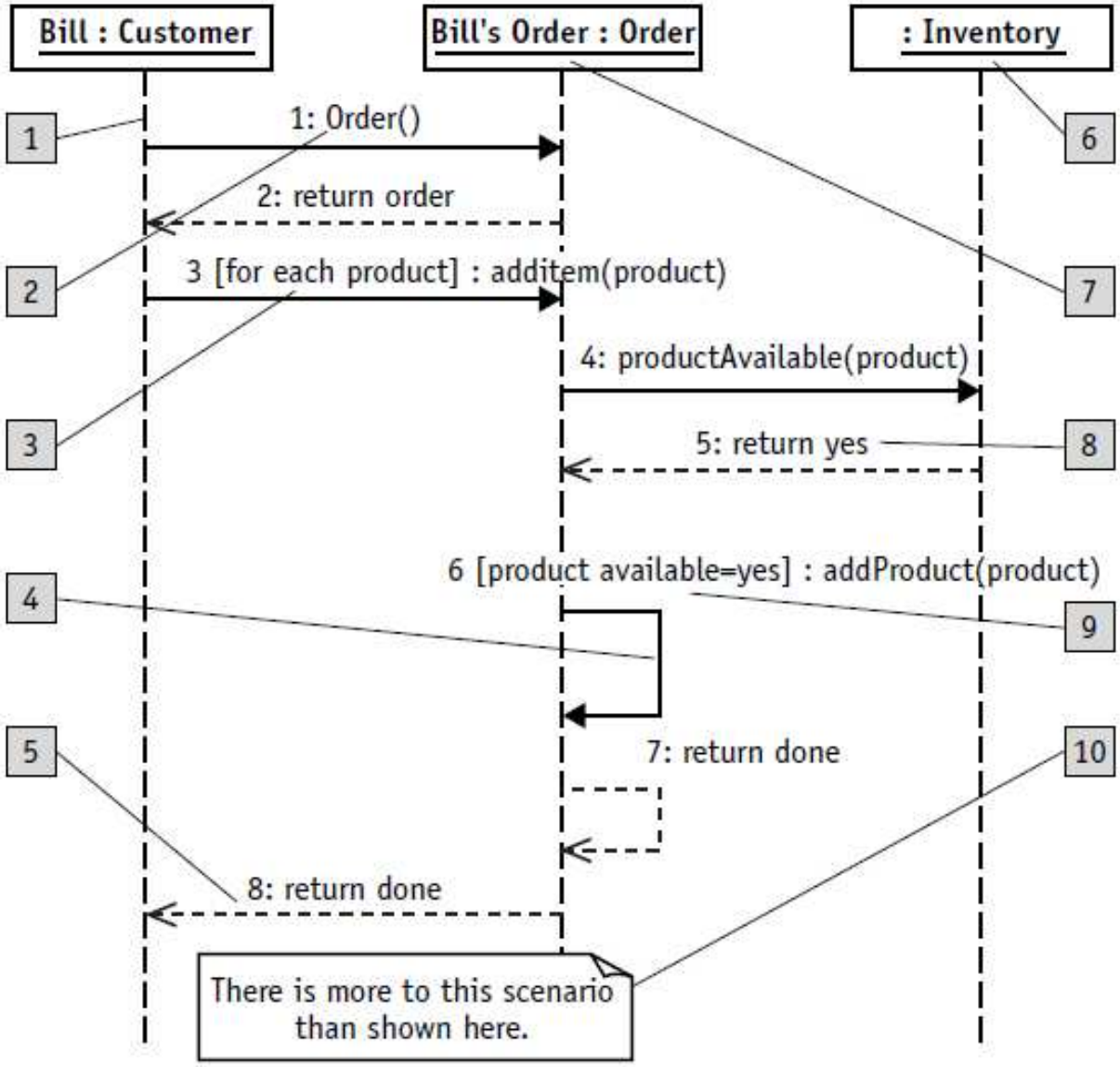
>> Building the Sequence diagram is easier if you have completed at least a first draft of the Use Case model and the Class diagram. From these two resources, you get sets of interactions (scenarios) and a pool of candidate objects to take responsibility for the interactions.

Elements of Sequence Diagram

>> The Sequence diagram uses three fundamental elements:

- participants/objects,
- messages/stimuli, and
- object lifeline.

Elements of Sequence Diagram



1. Object lifeline
2. Message/Stimulus
3. Iteration
4. Self-reference
5. Return
6. Anonymous object
7. Object name
8. Sequence number
9. Condition
10. Basic comment

Figure 16-2 Elements of Sequence diagram

Elements of Sequence Diagram

>> The **sequence numbers** are optional but are very helpful when you need to discuss the diagram and make changes.

>> Each **message** arrow describes an interface/operation on the object it is pointing to. Consequently, the message contains the operation signature, that is, the name, arguments, and optionally the return, such as addItem(product):boolean.

>> **Iteration** shows how you can indicate that an operation should be performed repeatedly. Use the square condition brackets to enclose either the number of times or the condition that controls the repetitions, for example [for each product].

>>You may use a UML **comment** to add information that is not explicitly part of the notation.

Extended Elements of Sequence Diagram

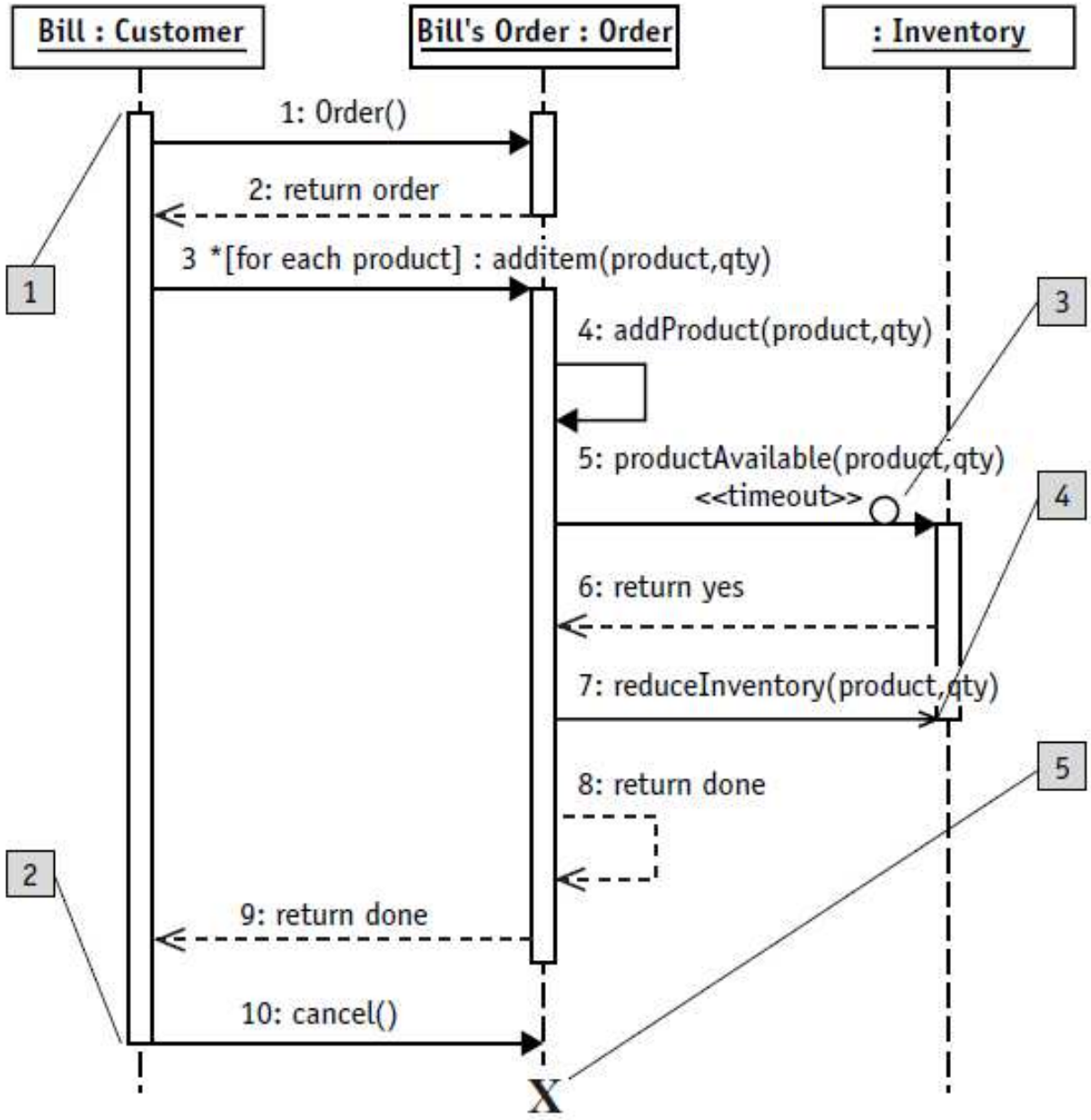


Figure 16-3 Extended elements of Sequence diagram

Extended Elements of Sequence Diagram

1. **Activation:** The start of the vertical rectangle, the activation bar

2. Deactivation: The end of the vertical rectangle, the activation bar

3. Timeout event: Typically signified by a full arrowhead with a small clock face or circle on the line

4. Asynchronous event: Typically signified by a stick arrowhead. There are times when the event is simply a signal to another object to do something and is not going to wait for response. An asynchronous message uses a stick arrowhead instead of the solid arrowhead used for simple or synchronous messages.

5. Object termination: symbolized by an X

Why We Need Sequence Diagram?

>> Sequence diagrams can be somewhat close to the code level. So why not just code up that algorithm rather than drawing it as a sequence diagram?

- a good sequence diagram is still a bit above the level of the real code (not EVERY line of code is drawn on diagram)

- sequence diagrams are language-independent (can be implemented in many different languages non-coders can do sequence diagrams

- easier to do sequence diagrams as a team

- can see many objects/classes at a time on same page

Exercises

Case 1:

In a withdrawal transaction of an ATM Machine system the customer inserts his ATM card in the machine. The machine then verifies the customer authentication using the information provided in customer account. After successful verification the machine takes withdrawal request from the customer and checks whether the request is valid or not. A valid request is carried out by the machine.

Exercises

Case 2:

In a library management system of a university a member can place a request to book a journal to the librarian. Before the librarian can complete the booking the member has to be verified of his status whether he is allowed to borrow journals or not. The journal then has to be located whether it is in the campus where the request was made or it is in a different campus. If the journal is in a different campus the librarian makes a request for the journal to be sent at the requested campus. The librarian then informs the member about the time required for the journal to reach and completes the booking.

Exercises

Case 3:

A doctor includes the instruction in the patient advice when a patient requires a bed or room in the hospital. The advice is then passed to the office clerk. The office clerk checks the present booking database to get the available room and bed list. Facilities of the rooms and the beds are then extracted from the room list. Available room and bed numbers and facilities are sent to the patient. The patient chooses a room or bed and then the office clerk writes patientid, room or bed no, doctorid and doctor advice in the booking database. Finally a copy of the admission is sent to the associated nurse and a copy is given to the patient.

Exercises

Case 4:

In a library management system of a student can borrow books. When a student brings the books he wants to borrow to the librarian, the librarian places a borrowing request by scanning the student ID. A borrowing object is created at that time which performs all the borrowing related tasks. It checks the student status from the database. If the student is a regular student and doesn't have any overdue book, the system allows for the books to be scanned. After the books are scanned the data is written in the database and the librarian is informed at the same time. If the student is regular but already has overdue books with him, the request is denied. If the student status is not regular, the request is also denied.

References

→ **Chapter 19**
The Unified Modeling Language User Guide
SECOND EDITION
By Grady Booch, James Rumbaugh, Ivar Jacobson

→ **Session 16 & 17**
UML Weekend Crash Course
Thomas A. Pender