THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

UML Diagrams

Slide Set - 9

Organized & Presented By:
Software Engineering Team CSED
TIET, Patiala

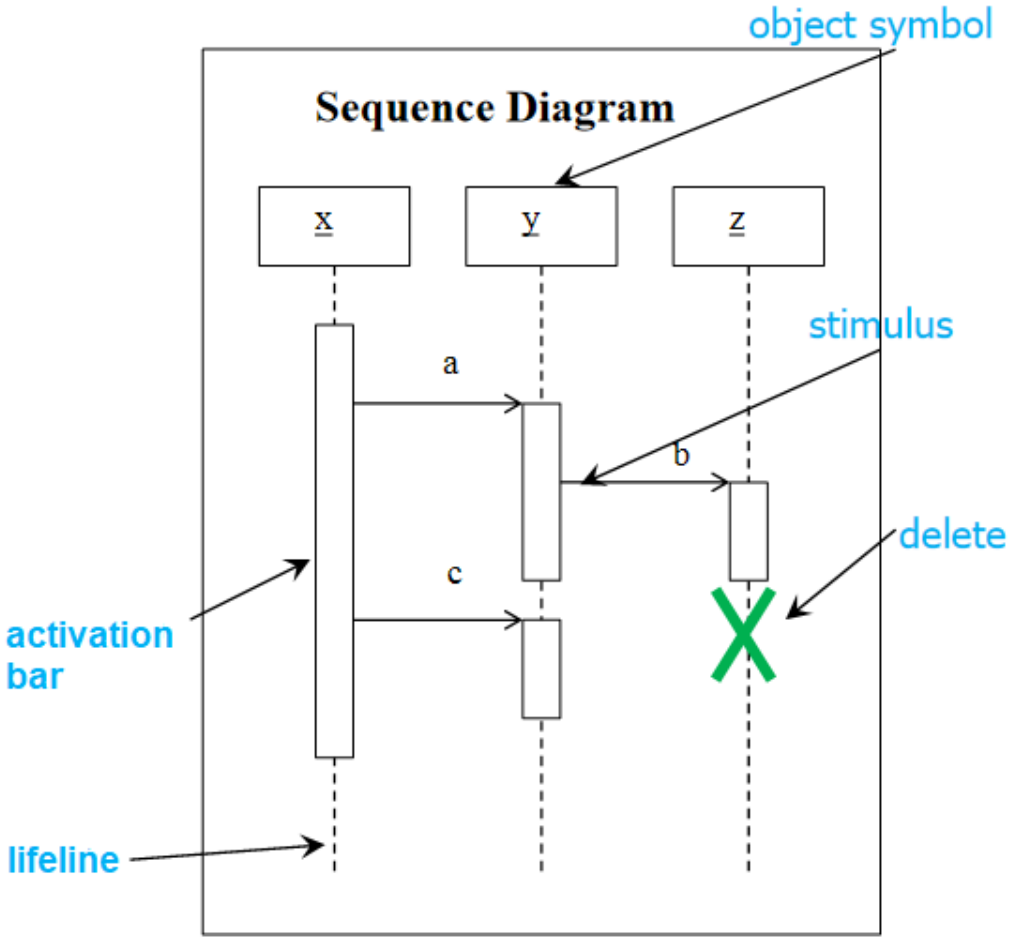
Sequence diagram

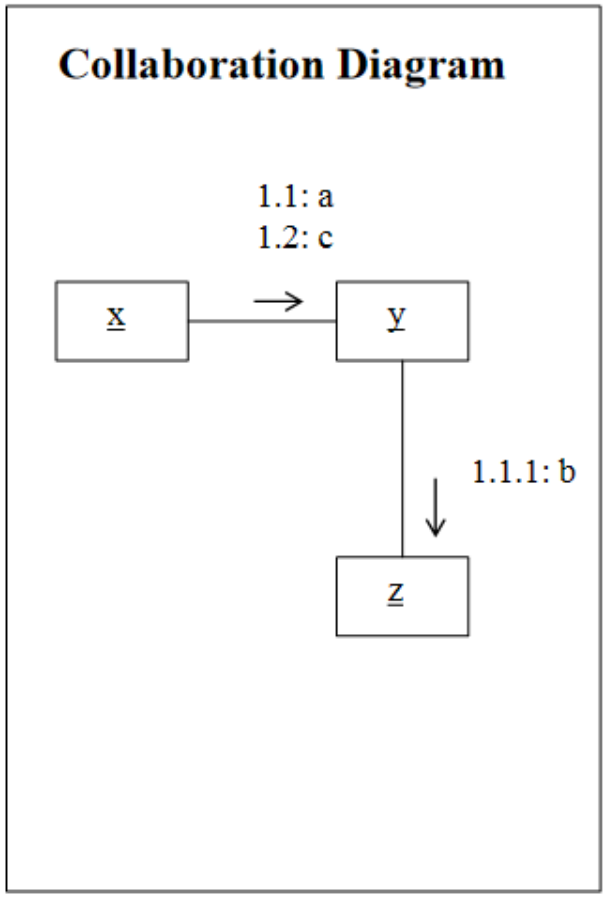
- Sequence diagrams demonstrate the behavior of objects in a use case by describing the objects and the messages they pass.
- Interaction: describes a collection of communications between instances, including all ways to affect instances, like operation invocation, as well as creation and destruction of instances
- Kinds
  - sequence diagram (temporal focus)
  - collaboration diagram (structural focus)

Sequence Diagram

- A sequence diagram is an interaction diagram that emphasizes the **time ordering of messages**.
- Sequence diagram **shows a set of objects and the messages sent and received by those objects.**
- The objects are typically named or anonymous instances of classes, but may also represent instances of other things, such as collaborations, components, and nodes.
- You use sequence diagrams to **illustrate the dynamic view of a system.**

Interaction Diagrams





Collaboration Diagram

- A collaboration diagram is an interaction diagram that **emphasizes the structural organization of the objects that send and receive messages.**

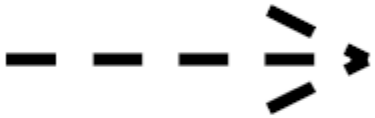
- Collaboration diagram shows a set of objects, links among those objects, and messages sent and received by those objects.

- The objects are typically named or anonymous instances of classes, but may also represent instances of other things, such as collaborations, components, and nodes.

- You use collaboration diagrams to **illustrate the dynamic view of a system**

Different Kinds of Arrows

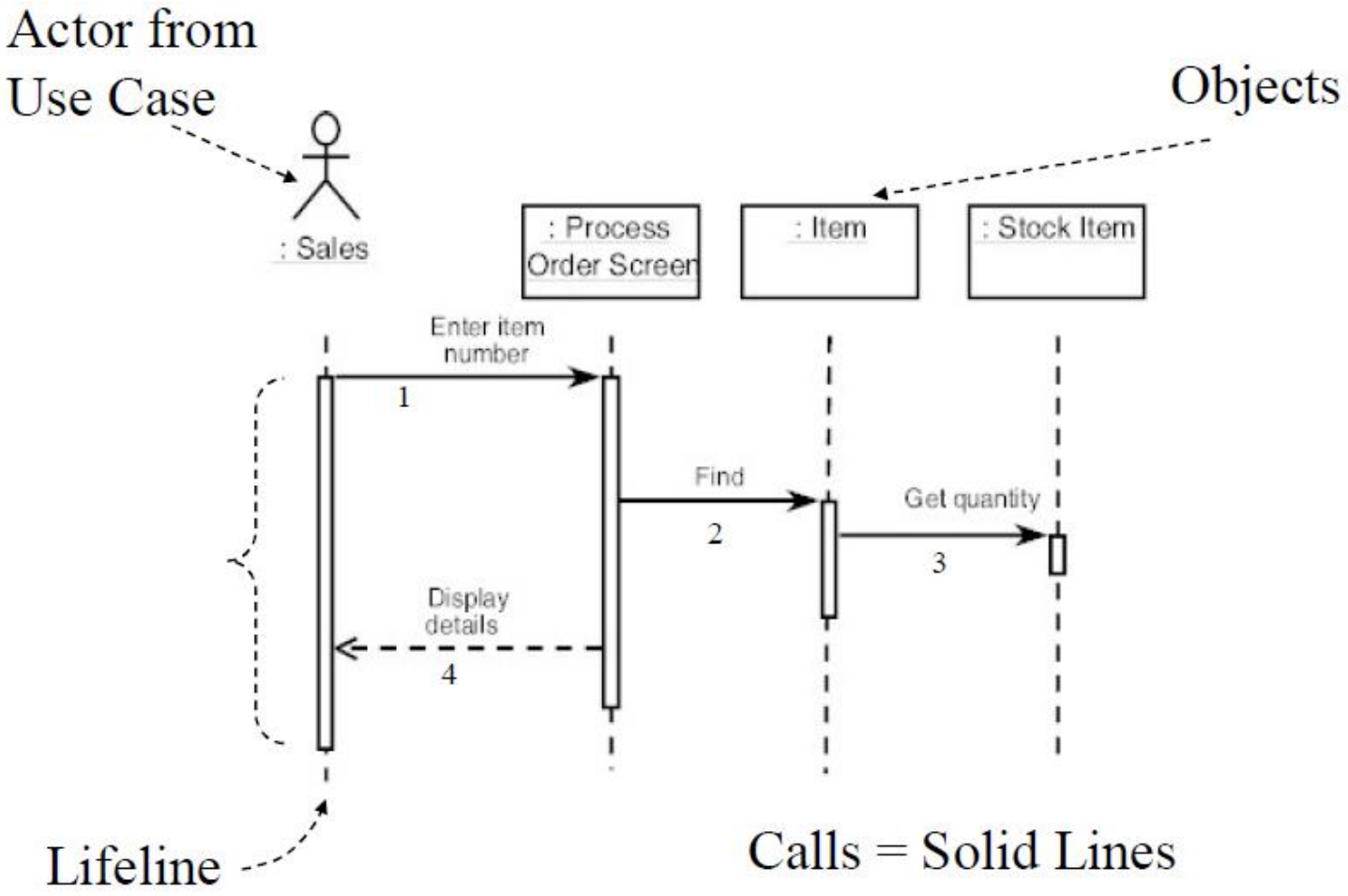
Procedure call or other kind of nested (synchronous) flow of control (caller waits for the callee to return)



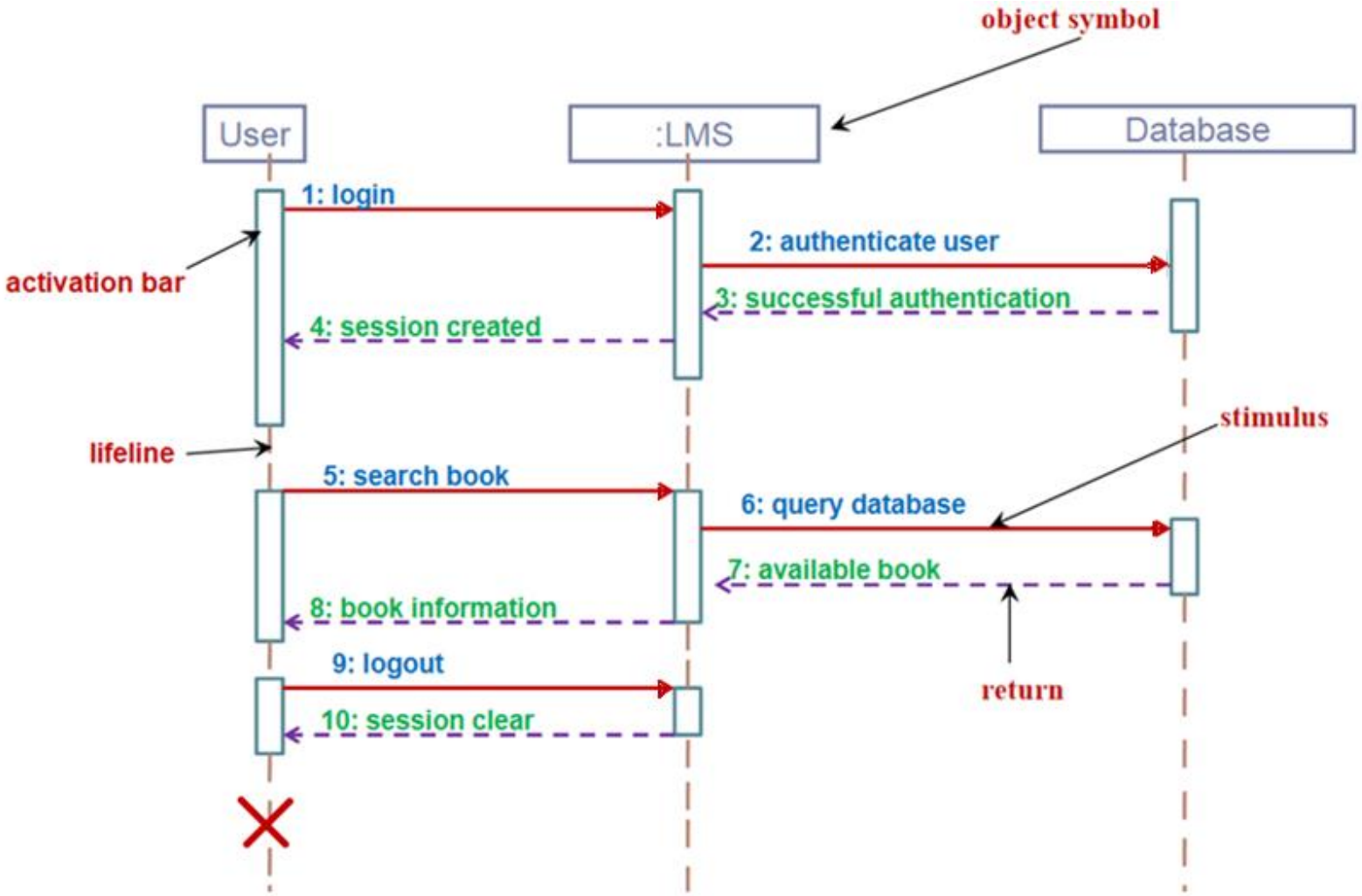
Return

Asynchronous flow of control (no waiting, no nesting, caller returns immediately)

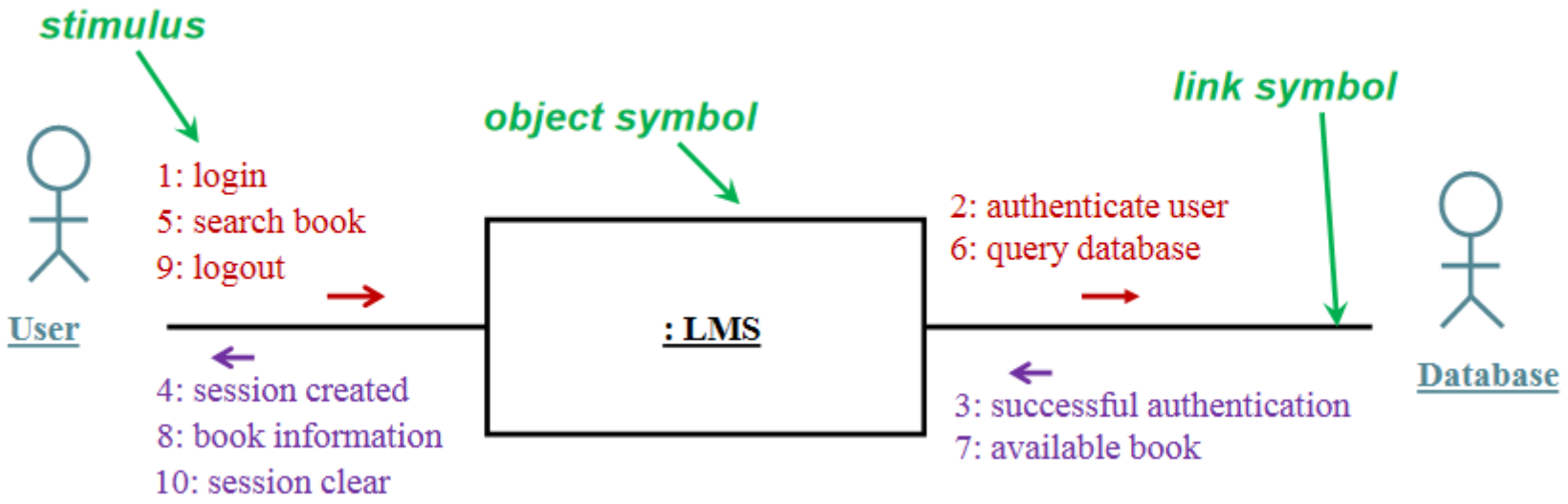
Example – Sequence Diagram



Sequence Diagram
Example – Library Management System



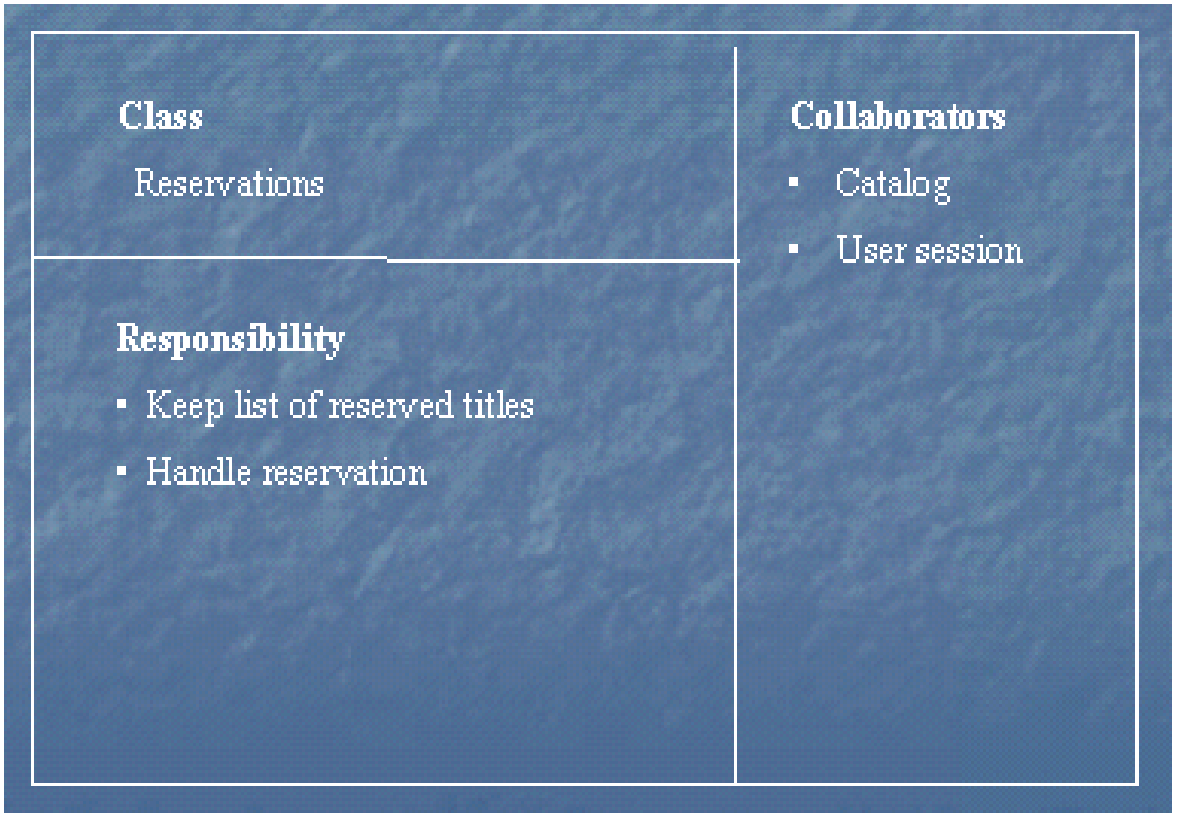
Collaboration Diagram
Example – Library Management System



Good Practice: CRC Card

(Class Responsibility Collaborator)

- Benefits: It is easy to describe how classes work by moving cards around; allows to quickly consider alternatives.



State Chart

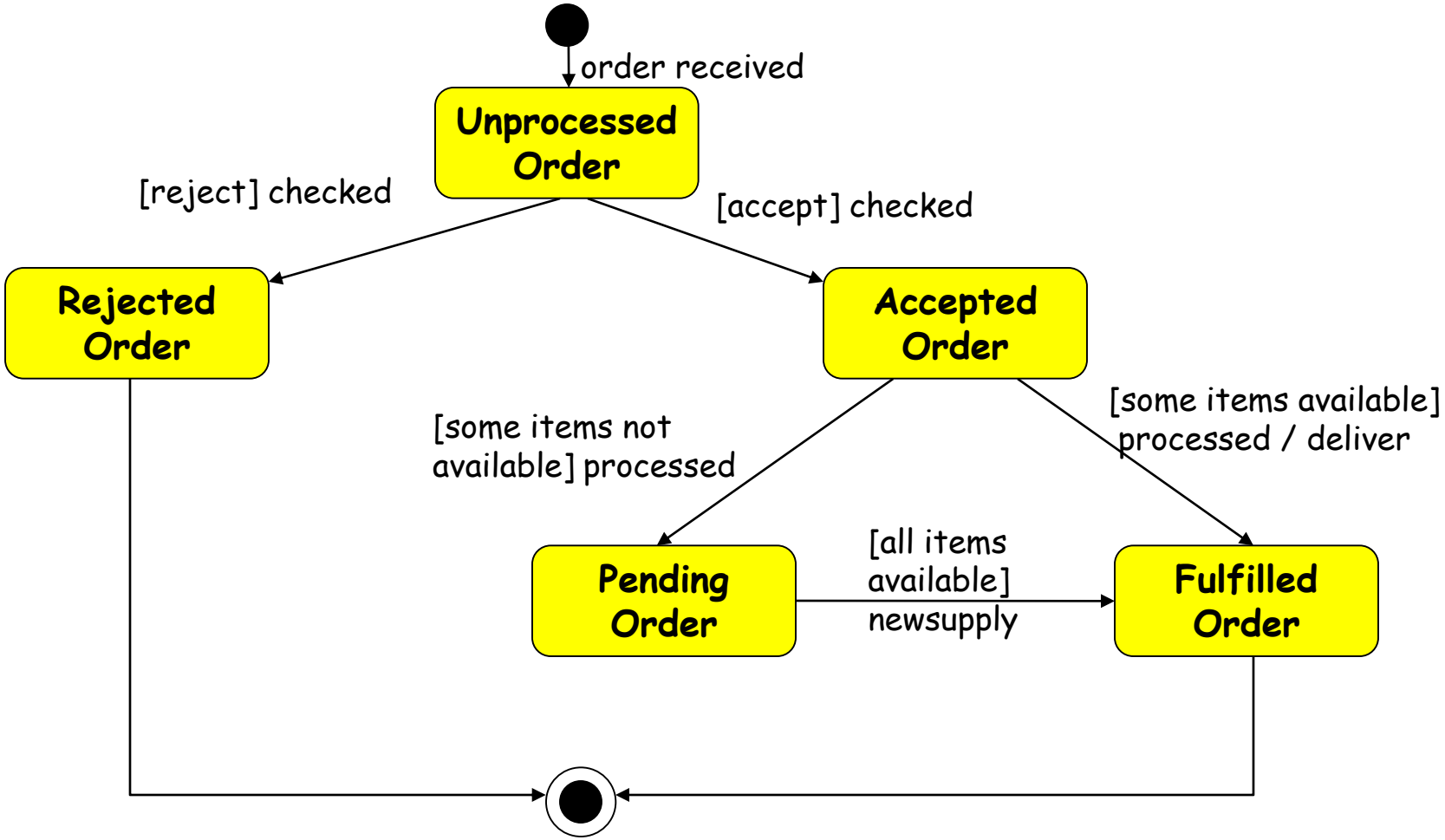
- It represents how an object goes through different states till the end of any process.
- It is based on the concept of finite state machines.

State Chart Diagram

– Elements of state chart diagram

State	Description	Symbol
Initial State	A filled circle	
Final State:	A filled circle inside a larger circle	
State	Rectangle with rounded corners	
Transitions	Arrow between states, also boolean logic condition (guard)	

An Example of A State Chart Diagram



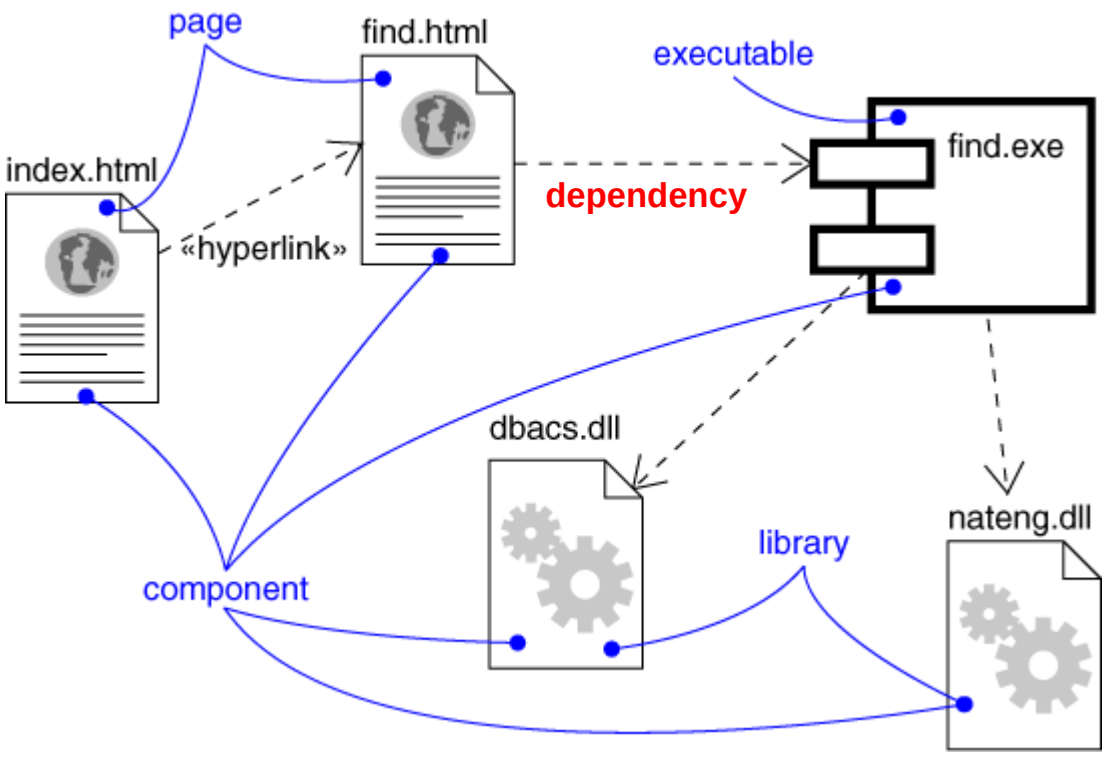
Example: State chart diagram for an order object

Component Diagram

- Captures the physical structure of the implementation
- Built as part of architectural specification
- Purpose
  - Organize source code
  - Construct an executable release
  - Specify a physical database
- Developed by architects and programmers

Component Diagram

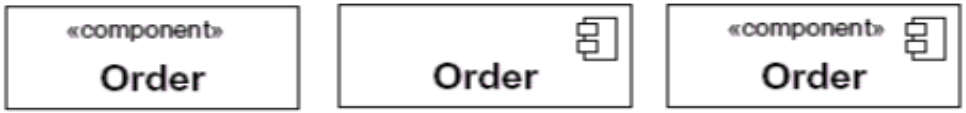
- Captures the physical structure of the implementation (code components)



Components:

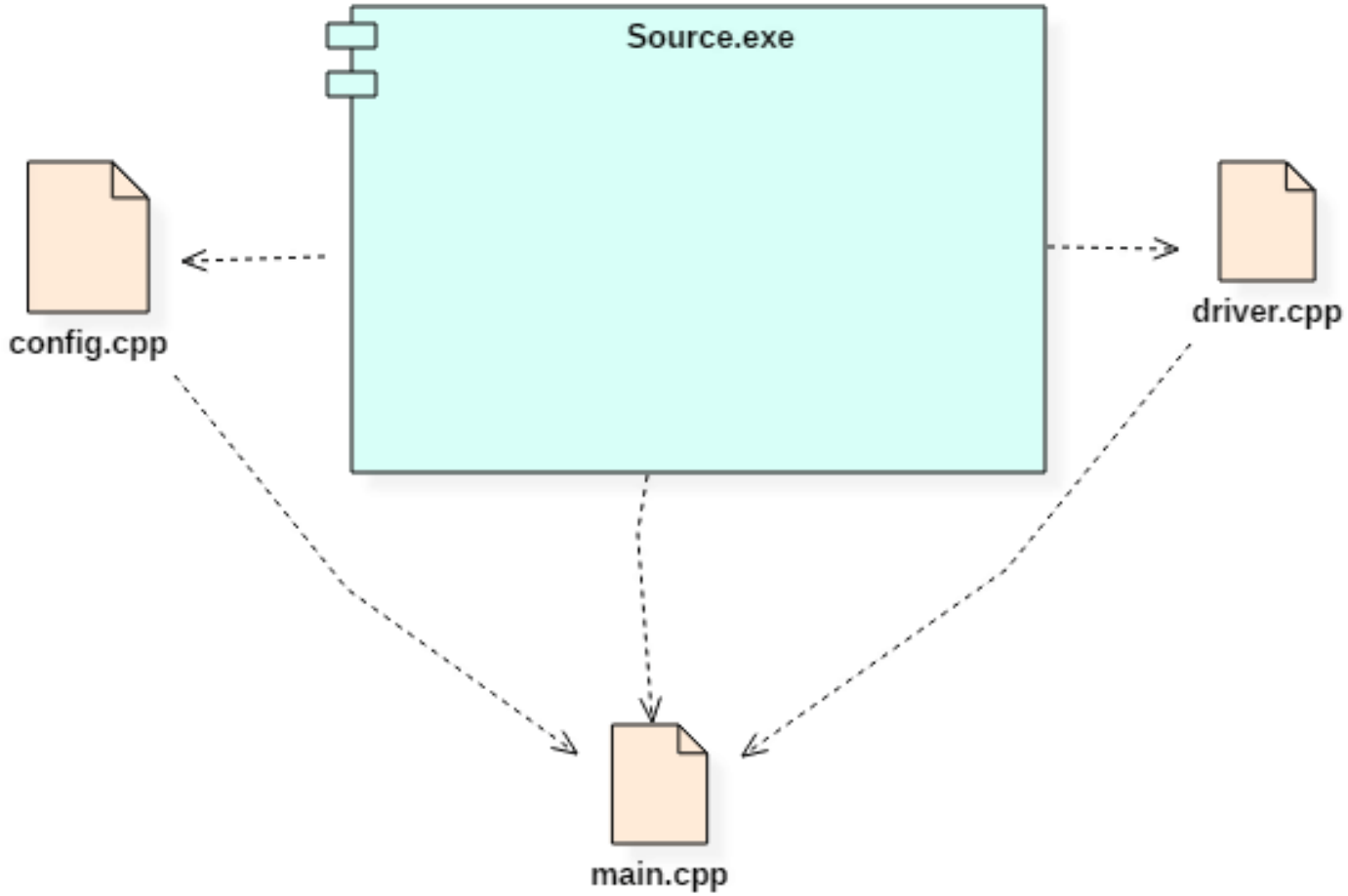
- **Executables**
- **Library**
- **Table**
- **File**
- **Document**

Basic Notations of Component Diagram



- In UML 2 specification, **"interface symbols"** connected to the outside of the rectangle.
- In this second approach the interface symbols with a complete circle at their end represent an interface that the component provides -- this lollipop" symbol is shorthand for a realization relationship. Interface symbols with only a **half circle** at their end represent **an interface that the component requires**. For example, the Order component *provides* two interfaces: OrderEntry and AccountPayable, and the Order component *requires* the Person interface.

Example – Component Diagram



Deployment Diagram

- Captures the topology of a system's hardware
- It includes:
  a) Hardware associated to the system
  b) Connections between the hardware components.

Deployment Diagram