Systems Analysis & Design

Dr. Arif Sari
Email: arif@arifsari.net

Course Website: www.arifsari.net/Courses/

Course Textbook:
Systems Analysis and Design With UML 2.0
An Object-Oriented Approach, Second Edition

Chapter 7:
Behavioural Modelling

Key Ideas

- Behavioral models describe the internal dynamic aspects of an information system that supports business processes in an organization
- Key UML behavioral models are: sequence diagrams and behavioural state machines

Objectives

- Understand the rules and style guidelines for sequence diagrams and behavioral state machines.
- Understand the processes used to create sequence diagrams and behavioral state machines.
- Be able to create sequence diagrams and behavioral state machines.
- Understand the relationship between the behavioral models and the structural and functional models.

BEHAVIORAL MODELS

Purpose of Behavioral Models

- Show how objects collaborate to support each use case in the structural model
- Depict the internal view of the business process
- To show the effects of varied processes on the system

Interaction Diagram Components

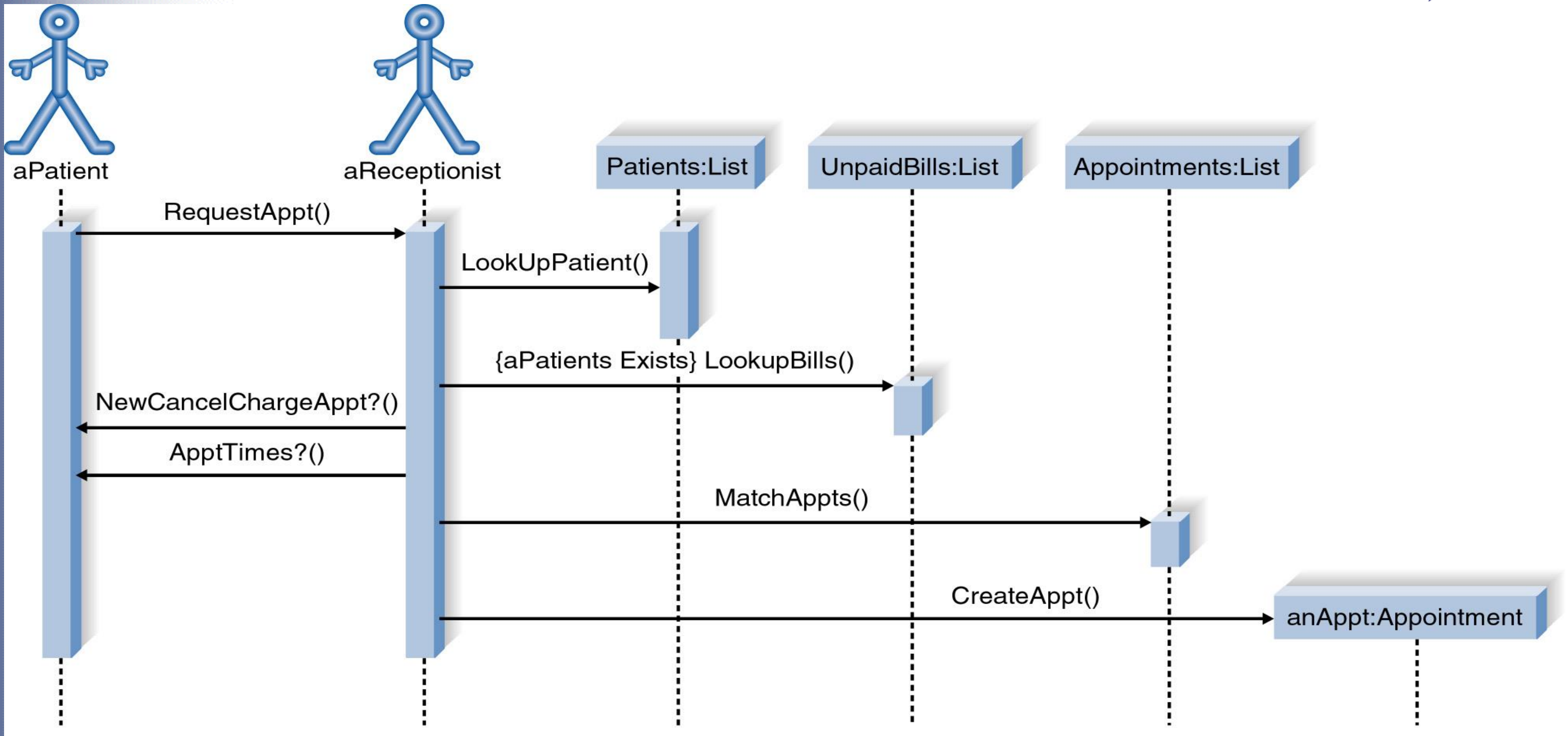
- **Objects**
  - Instance of a class
- **Operations**
  - Send and receive messages
- **Messages**
  - Tell object to execute a behavior

WILEY

Sequence Diagrams

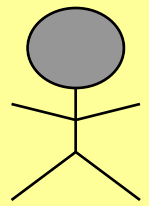
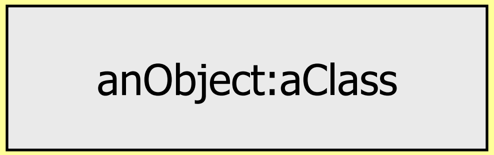
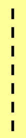
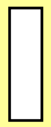
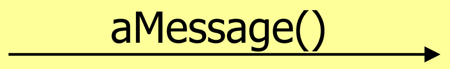
- Illustrate the **objects** that participate in a use-case
- Show the **messages** that pass between objects **for a particular use-case**

Example Sequence Diagram Make Appointment



Dennis: SAD
Fig: 8-1 W-30 100% of size
Fine Line Illustrations (516) 501-0400

Sequence Diagram Syntax

ACTOR	
OBJECT	
LIFELINE	
EXECUTION OCCURRENCE (FOCUS OF CONTROL)	
MESSAGE	
OBJECT DESTRUCTION	X

Building a Sequence Diagram

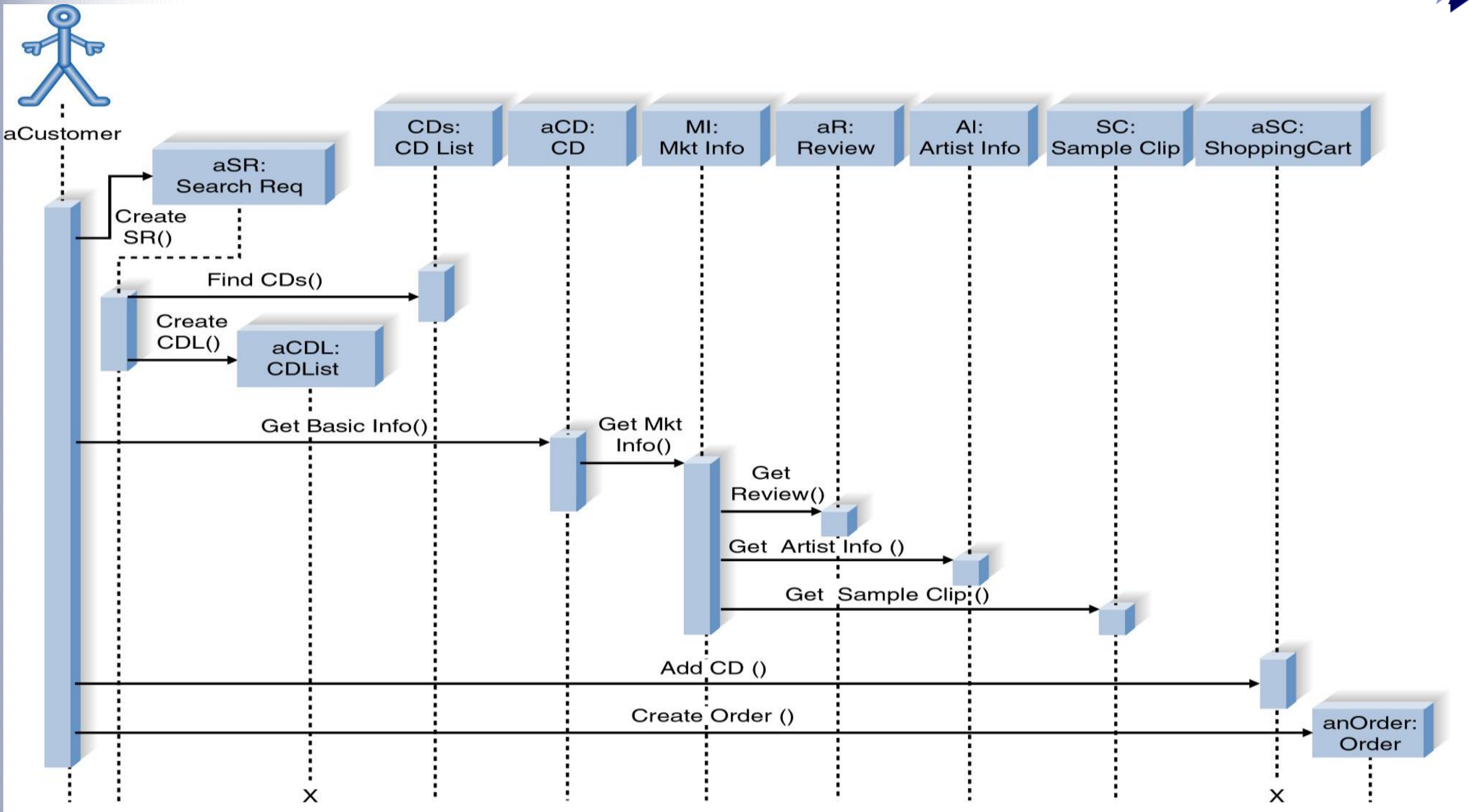
1. Determine the **context** of the sequence diagram
2. Identify the participating **objects**
3. Set the **lifeline for each object**
4. Add **messages**
5. Place the **execution occurrence (focus of control)** on each object's lifeline
6. **Validate** the sequence diagram

CD Selections

Normal Flow of Events:

1. Customer submits a search request to the system.
2. The system provides the customer a list of recommended CDs.
3. The customer chooses one of the CDs to find additional information.
4. The system provides the customer with basic information & CD Reviews
5. The customer calls the maintain order use case.
6. The customer iterates over 3 through 5 until finished shopping.
7. The customer executes the checkout use case.
8. The customer leaves the website.

CD Selections



Dennis: SAD
Fig: 8-5 W-32 100% of size
Fine Line Illustrations (516) 501-0400

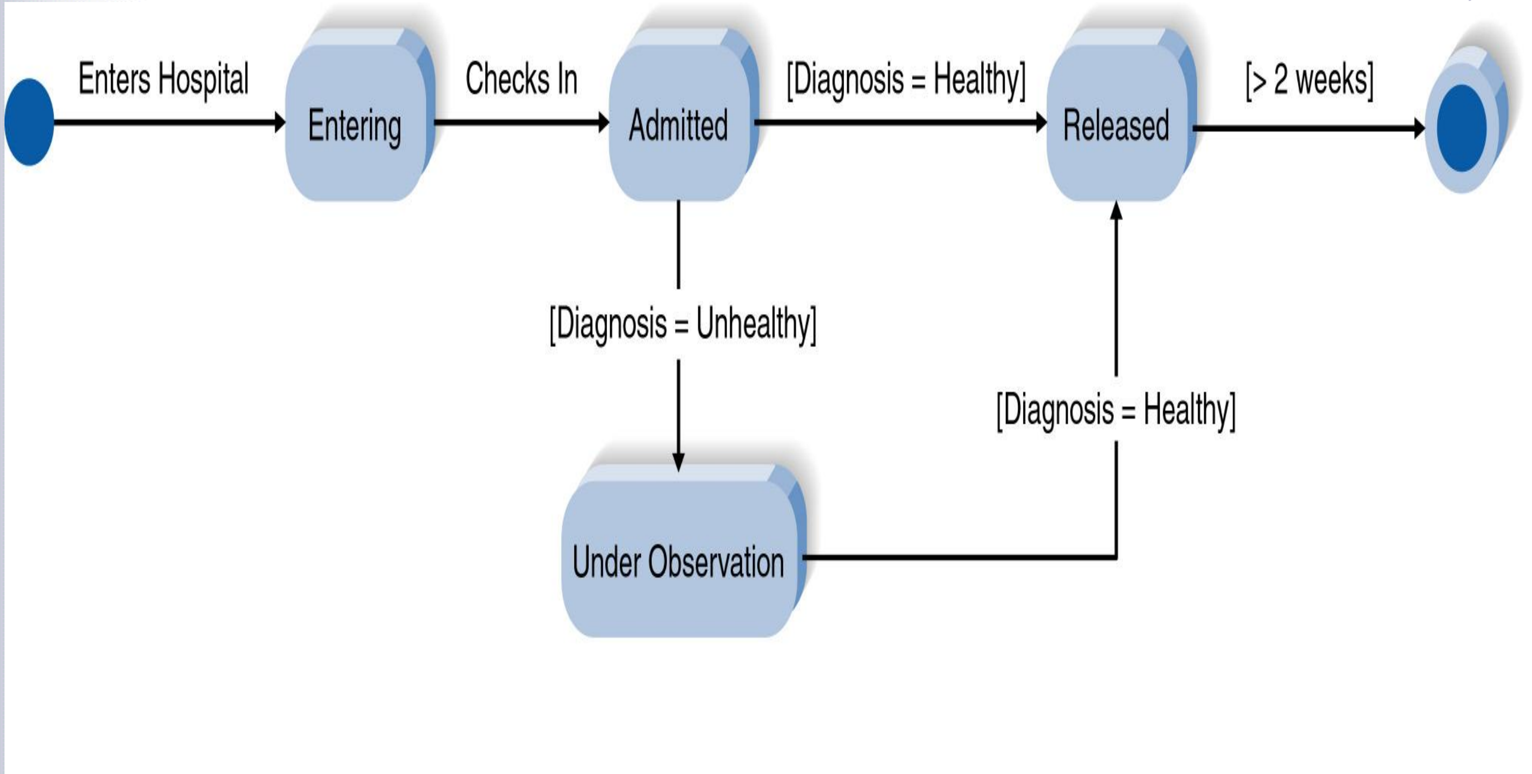
Behavioral State Machines (State Chart Diagrams)

- The behavioral state machine is a **dynamic** model that shows the different **states** of the object and what **events** cause the object to change from one state to another, along with its responses and actions.

Elements of a Behavioral State Machine

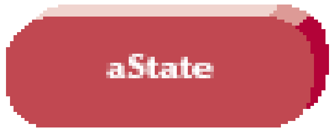
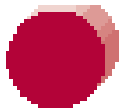
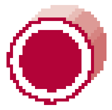
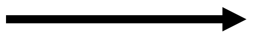
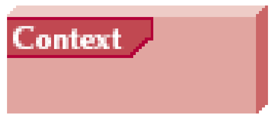
- States (idle conditions)
- Events (triggers)
- Transitions (changes in state)
- Actions (cause transitions)
- Activities (groups of actions)

Example Behavioral State Machine Diagram



Dennis: SAD
Fig: 8-11 W-37 100% of size
Fine Line Illustrations (516) 501-0400

Behavioral State Machine Diagram Syntax

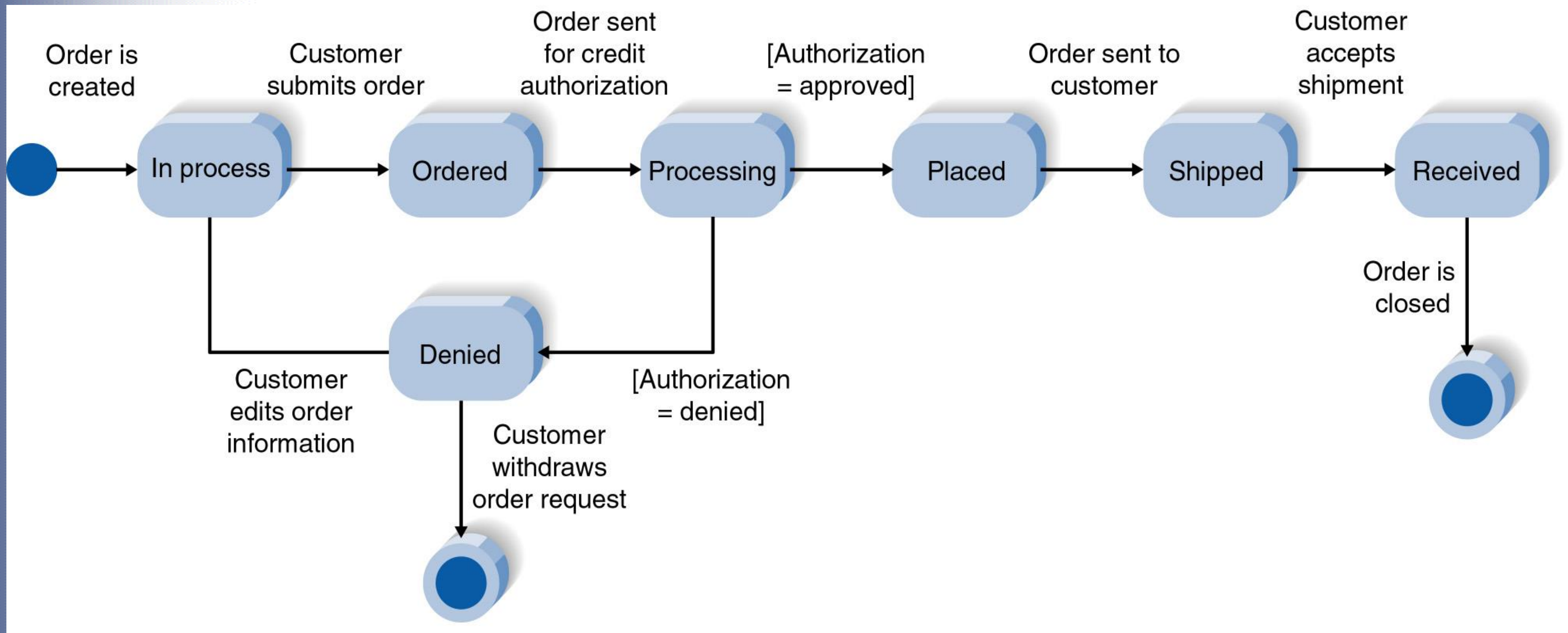
A STATE	
AN INITIAL STATE	
A FINAL STATE	
AN EVENT	anEvent
A TRANSITION	
A Frame	

WILEY

Building Behavioral State Machine Diagrams

- Set the context
- Identify the initial final, and stable states of the object
- Determine the order in which the object will pass through stable states
- Identify the events, actions, and guard conditions associated with the transitions
- Validate the state machine diagram

CD Selections



Dennis: SAD
Fig: 8-15 W-38a 100% of size
Fine Line Illustrations (516) 501-0400

Summary

- *Sequence diagrams* illustrate the classes that participate in a use case and the messages that pass between them.
- *Behavioral State Machine diagrams* show the different states that a single class passes through in response to events.