Chapter 8: Behavioral Modeling

John Wiley & Sons, Inc.
Copyright 2005
Copyright © 2005
John Wiley & Sons, Inc.

- All rights reserved. Reproduction or translation of this work beyond that permitted in Section 117 of the 1976 United States Copyright Act without the express written permission of the copyright owner is unlawful.
- Request for further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.
- The purchaser may make back-up copies for his/her own use only and not for redistribution or resale.
- The Publisher assumes no responsibility for errors, omissions, or damages, caused by the use of these programs or from the use of the information contained herein.
Behavioral Modeling

Chapter 8
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, collaboration diagrams, and statechart diagrams.
Objectives

- Understand the rules and style guidelines for sequence and communication diagrams and behavioral state machines.
- Understand the processes used to create sequence and communication diagrams and behavioral state machines.
- Be able to create sequence and communication 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**
  - Instantiation of a class
  - Has attributes that describe an object

- **Operations**
  - Send and receive messages

- **Messages**
  - Tell object to execute a behavior
Sequence Diagrams

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

- **AN ACTOR**

- **AN OBJECT**
  - `anObject: aClass`

- **A LIFELINE**

- **A FOCUS OF CONTROL**

- **A MESSAGE**
  - `aMessage()`

- **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 focus of control on each object’s lifeline
6. Validate the sequence diagram
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

Slide 14

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

- Essentially an object diagram that shows message passing relationships instead of aggregation or generalization associations.
- Emphasize the flow of messages among objects, rather than timing and ordering of messages.
Example Collaboration Diagram

1: RequestAppt()

4: NewCancelChangeAppt?(())

5: ApptTimes?()

2: LookUpPatient()

3: [aPatient Exists] LookupBills()

6: MatchApps()

7: CreateAppt()

PATIENTS:LIST

UNPAIDBILLS:LIST

APPOINTMENTS:LIST

ANAPPT:APPOINTMENT

Dennis: SAD
Fig. 8-6  W-33  100% of size
Fine Line Illustrations (516) 501-0400
<table>
<thead>
<tr>
<th>Actor</th>
<th>![anActor]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>![anObject: aClass]</td>
</tr>
<tr>
<td>Association</td>
<td>![1: a Message]</td>
</tr>
<tr>
<td>Message</td>
<td>![Context]</td>
</tr>
<tr>
<td>Frame</td>
<td></td>
</tr>
</tbody>
</table>
Communication Diagrams

1. Set the context.
2. Identify which objects (actors) and the associations between the objects participate in the collaboration.
3. Layout the communication diagram.
4. Add messages.
5. Validate the communication diagram.
"CRUD" Analysis Example

<table>
<thead>
<tr>
<th>Customer</th>
<th>SearchReq</th>
<th>CDList</th>
<th>CD</th>
<th>Mkt Info</th>
<th>Review</th>
<th>Artist Info</th>
<th>Sample Clip</th>
<th>Shopping Cart</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SearchReq</td>
<td>CR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDList</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mkt Info</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>Review</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Artist Info</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample Clip</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shopping Cart</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Order</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 8.9** CRUD Matrix for the Place Order Use Case
Building a Collaboration Diagram

- Determine the context of the collaboration diagram
- Identify the participating objects and their associations
- Layout objects and associations
- Add messages
- Validate the sequence diagram
Behavioral State Machines

- 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
- Events
- Transitions
- Actions
- Activities
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

<table>
<thead>
<tr>
<th>Term</th>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>A STATE</td>
<td>![aState]</td>
</tr>
<tr>
<td>AN INITIAL STATE</td>
<td>![anInit]</td>
</tr>
<tr>
<td>A FINAL STATE</td>
<td>![aFinal]</td>
</tr>
<tr>
<td>AN EVENT</td>
<td>![anEvent]</td>
</tr>
<tr>
<td>A TRANSITION</td>
<td><img src="arrow" alt="" /></td>
</tr>
<tr>
<td>A Frame</td>
<td>![Context]</td>
</tr>
</tbody>
</table>
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 statechart diagram
Your Turn

- What distinguishes the sequence diagram, the collaboration diagram, and the behavioral state machine diagram?
- For what sort of new applications might you need to develop all of these? Are there any new applications that would not need all of these diagrams for full development?
Summary

- **Sequence diagrams** illustrate the classes that participate in a use case and the messages that pass between them.
- **Collaboration diagrams** provide a dynamic view of the object-oriented system and accentuate message passing between collaborating actors and objects.
- **Behavioral State Machine diagrams** show the different states that a single class passes through in response to events.
Expanding the Domain

- Each year the Association for Computing Machinery (ACM) sponsors a conference on object oriented programming. For details about future conferences and other ACM programs check:
  - http://oosplsla.acm.org
  - http://www.acm.org