

SWE 621: Software Modeling and Architectural Design

Lecture Notes on Software Design

Lecture 6 – Dynamic Modeling

Hassan Gomaa
Dept of Computer Science
George Mason University
Fairfax, VA

Copyright © 2011 Hassan Gomaa

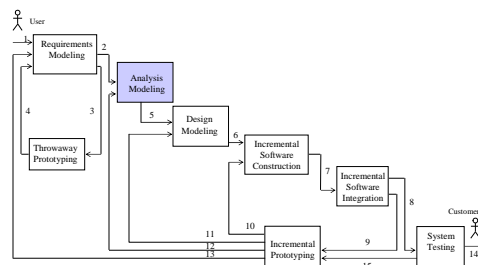
All rights reserved. No part of this document may be reproduced in any form or by any means, without the prior written permission of the author.

This electronic course material may not be distributed by e-mail or posted on any other World Wide Web site without the prior written permission of the author.

Copyright 2011 H. Gomaa

Steps in Using COMET/UML

- 1 Develop Software Requirements Model
- 2 **Develop Software Analysis Model**
 - Develop static model of problem domain (Chapter 7)
 - Structure system into objects (Chapter 8)
 - Develop statecharts for state dependent objects (Chapter 10)
 - **Develop object interaction diagrams for each use case (Chapter 9, 11)**
- 3 Develop Software Design Model



Copyright 2011 H. Gomaa

2

Dynamic Interaction Modeling

Section 7

Hassan Gomaa

Reference: H. Gomaa, Chapters 9,11 - *Software Modeling and Design*, Cambridge University Press, February 2011

H. Gomaa, Chapter 11 - *Designing Concurrent, Distributed, and Real-Time Applications with UML*, Addison Wesley Object Technology Series, July, 2000

Copyright © 2011 Hassan Gomaa

All rights reserved. No part of this document may be reproduced in any form or by any means, without the prior written permission of the author.

Copyright 2011 H. Gomaa

3

Dynamic Modeling

- Use cases realized in Dynamic Model
 - Show objects participating in each use case
- Determine how objects participate in use case
 - Use object structuring criteria to determine objects
 - Stereotype for each object structuring criterion
 - Shows sequence of object interactions in use case
 - Depict on
 - **communication diagram** or
 - **sequence diagram**
- State dependent control objects
 - Modeled using statecharts
- Dynamic Modeling
 - Approach to determine how objects interact with each other to support use case

Copyright 2011 H. Gomaa

4

Dynamic Modeling

- Determine how objects interact with each other to support use case
 - Start with external event from actor
 - Determine objects needed to support use case
 - Determine sequence of internal events following external event
 - Depict on communication diagram or sequence diagram
- Stateless Dynamic Modeling
- State dependent Dynamic Modeling
 - Important for real-time design

Copyright 2011 H. Gomaa

5

Communication Diagram

- Graphically depicts objects interacting with each other
 - Show objects as boxes
 - Show their message interactions as arrows
 - Number sequence of messages

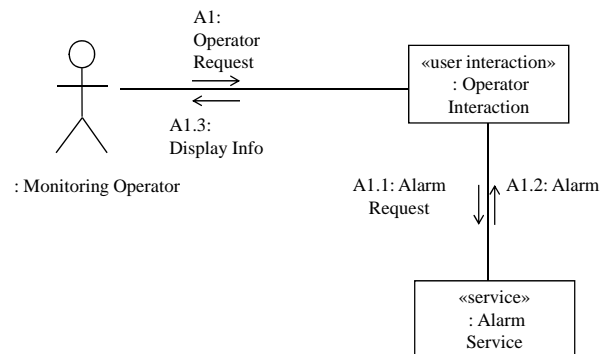


Figure 9.2 Communication diagram for View Alarms use case
Copyright 2011 H. Gomaa

6

Sequence Diagram

- Shows sequence of object interactions in use case
- Emphasis on messages passed between objects
 - Objects represented by vertical lines
 - Messages represented by labeled horizontal arrows

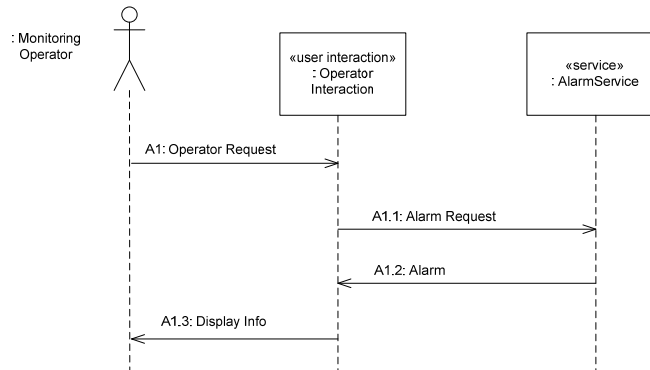


Figure 9.3 Sequence diagram for View Alarms use case

Copyright 2011 H. Gomma

7

Dynamic Modeling

- Determine how objects interact with each other to support use case
 - Start with external event from actor
 - Determine objects needed to support use case
 - Determine sequence of internal events following external event
 - Depict on communication diagram or sequence diagram
- Stateless (non state-dependent) Dynamic Modeling
- State dependent Dynamic Modeling

Copyright 2011 H. Gomma

dm-8

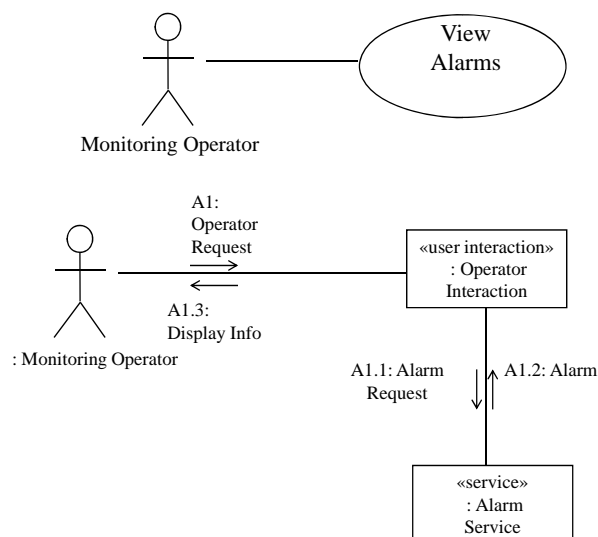
Stateless Dynamic Modeling

- Start with use case
- Determine boundary objects
 - Receives external events from actor
- Determine internal objects
 - Receive messages from boundary objects
- Determine object interactions
 - Sequence of messages passed
- Develop main interaction sequence (scenario)
- Develop alternative sequences
 - For alternative branches of use case
 - E.g, for error handling or less frequently occurring conditions

Copyright 2011 H. Gomma

dm-9

Example of Stateless Dynamic Modeling
Figure 9.2 Communication diagram for View Alarms use case



Copyright 2011 H. Gomma

dm-10

Message Sequence Description

- Describes sequence of object interactions
 - Narrative description
 - Corresponds to Communication Diagram or Sequence Diagram
- Description corresponds to message sequence numbering on diagrams
 - Describe what object does on receiving message
 - E.g., every time an object interacts with an entity object
 - Describe the object being accessed
 - Identify attributes referenced

Message Sequence Description - View Alarms

A1: The operator requests to view one or more alarms—for example, to view the status of a factory workstation.

A1.1: Operator Interaction object sends an alarm request to the Alarm Service.

A1.2: Alarm Service responds—for example, with information about the requested alarm: name, description, location, severity.

A1.3: Operator Interaction object formats and displays the alarm information to the operator.

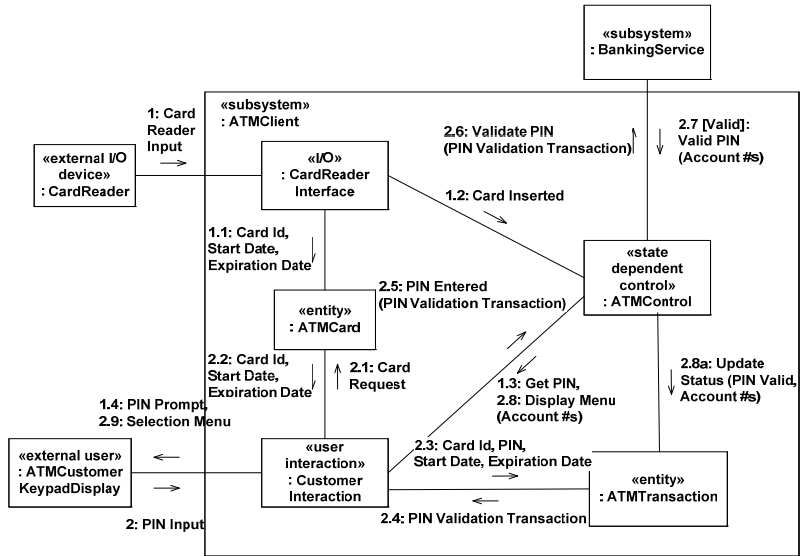
State Dependent Dynamic Modeling

- Object interaction controlled by statechart(s)
 - Control object
 - Executes statechart
 - Activates/deactivates other objects
- For each use case
 - Determine objects participating in use case
 - Determine sequence of object communication
 - Develop statechart for control object
- For each event that arrives at control object
 - Determine state transition from current state to next state
 - Determine actions and activities to be executed
 - Determine objects required to perform actions and activities

Example of Dynamic Modeling Banking System - Validate PIN Use Case

- ATM Client Objects
 - Boundary Objects
 - Card Reader Interface
 - Customer Interaction
 - Entity Objects
 - ATM Card
 - ATM Transaction
 - State Dependent Control Object
 - ATM Control

Figure 11.1 Communication diagram: ATM Client – Validate PIN use case – Valid Pin

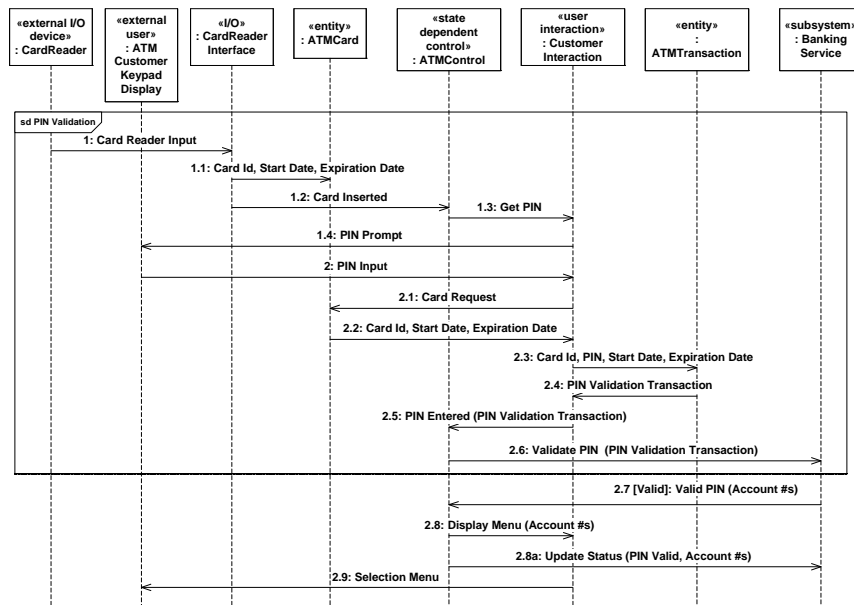


PIN Validation Transaction = {transactionId, transactionType, cardId, PIN, startDate, expirationDate}

Copyright 2011 H. Gomma

15

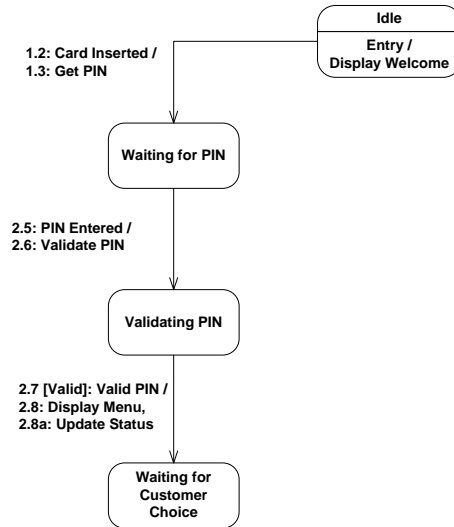
Figure 11.2 Sequence Diagram for Validate PIN use case – Valid Pin



Copyright 2011 H. Gomma

16

Figure 11.3 Validate PIN Statechart – Valid Pin



Copyright 2011 H. Goma

17

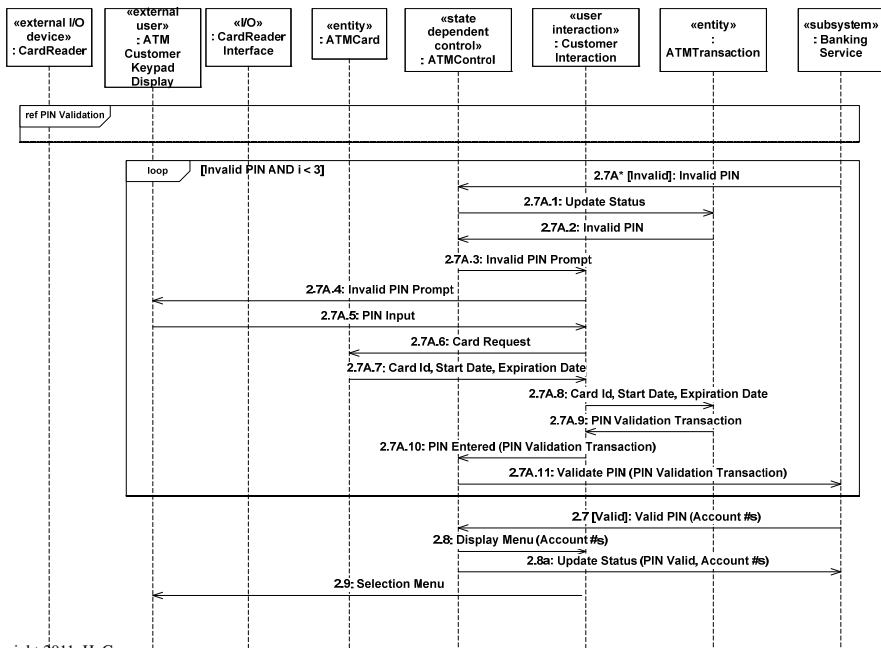
Validate PIN Main and Alternative Scenarios

- **Valid card and PIN entered**
2.6 [Valid]: Valid PIN
- **Incorrect PIN**
2.6A* [Invalid]: Invalid PIN
Iteration of messages: 2.6A.1 through 2.6A.8.
- **Incorrect PIN entered three times**
2.6B [Third Invalid]: Third Invalid PIN
- **Card stolen or expiration date expired.**
2.6C [Stolen OR expired]: Card stolen, Card expired
- **Card not recognized:**
1.1A [card not recognized] Eject Card
- **Cancel**
2A.2 Eject

Copyright 2011 H. Goma

dm-18

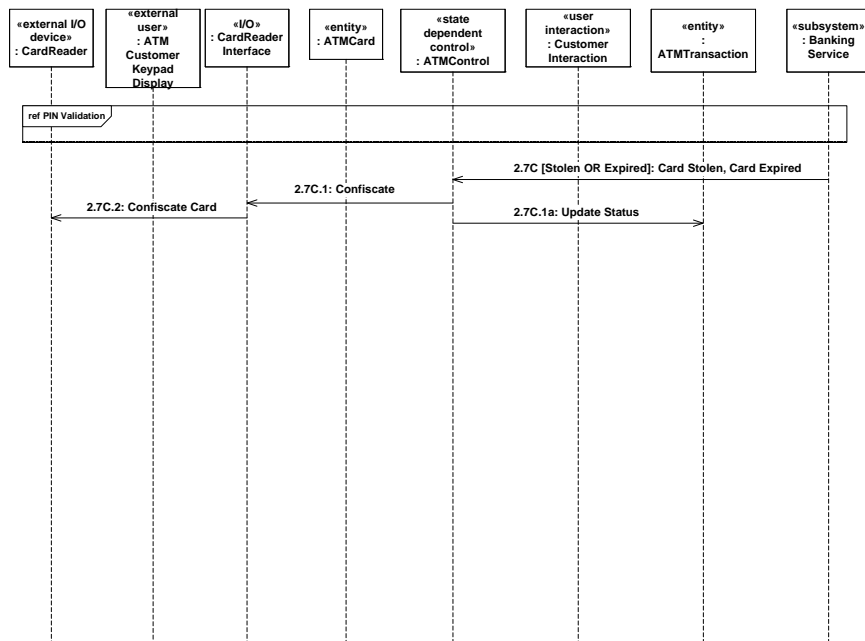
Figure 11.4 Sequence Diagram for Validate PIN use case – Invalid Pin



Copyright 2011 H. Gomma

19

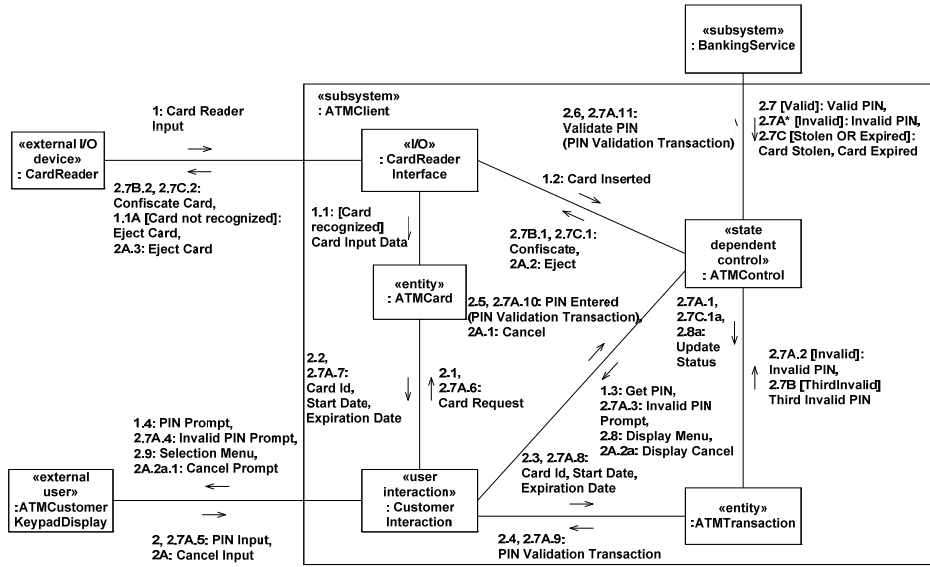
Figure 11.6 Sequence Diagram for Validate PIN use case – Stolen or Expired Card



Copyright 2011 H. Gomma

20

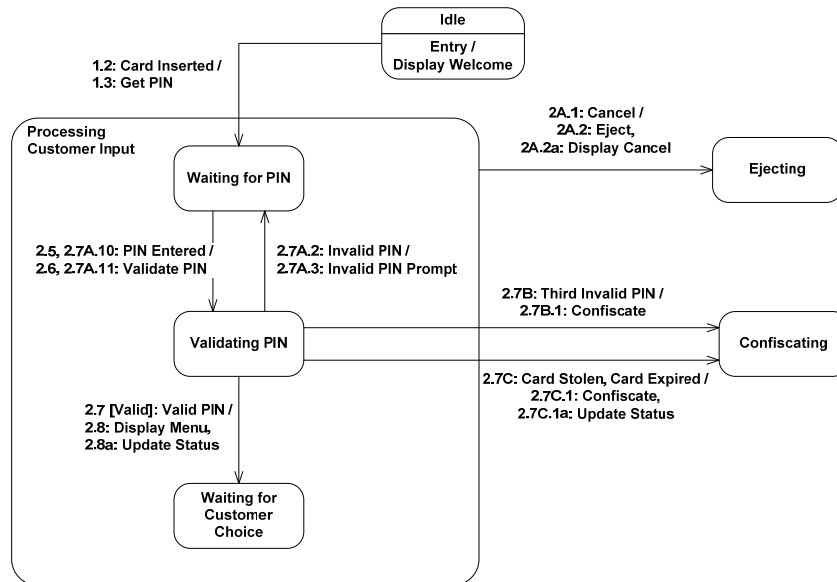
Figure 11.7 Communication diagram: ATM Client – Validate PIN use case – Generic form showing alternatives



Copyright 2011 H. Gomaa

21

Figure 11.8 Validate PIN Statechart – Valid Pin use case showing alternatives



Copyright 2011 H. Gomaa

22

Example of Dynamic Modeling Banking System - Validate PIN Use Case

- Banking Service Objects
 - Entity Objects
 - Debit Card
 - Card Account
 - Transaction Log
 - Business Logic Objects
 - PIN Validation Transaction Manager
- Example: Fig. 21.14, 21.15

Copyright 2011 H. Gomma

dm-23

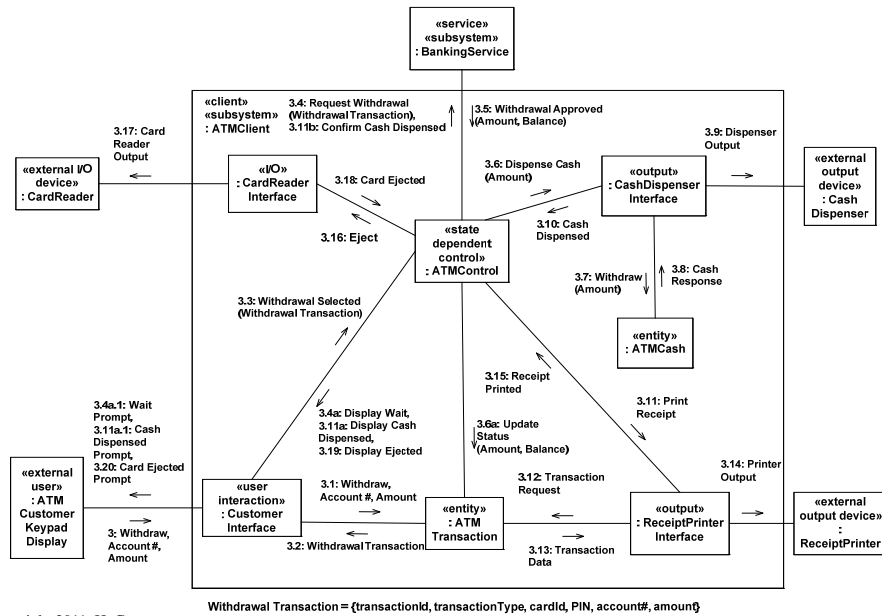
Example of Dynamic Modeling Banking System - Withdraw Funds Use Case

- ATM Client Objects
 - Boundary Objects
 - Card Reader Interface
 - Receipt Printer Interface
 - Cash Dispenser Interface
 - Customer Interaction
 - Entity Objects
 - ATM Transaction
 - ATM Cash
 - State Dependent Control Object
 - ATM Control
- Figs. 21.16- 21.18

Copyright 2011 H. Gomma

dm-24

Figure 21.16 Communication diagram: ATM Client – Withdraw Funds use case



Copyright 2011 H. Gomma

25

Example of Dynamic Modeling Banking System - Withdraw Funds Use Case

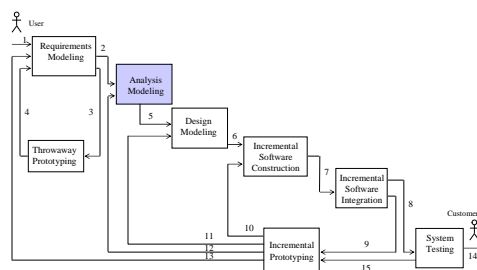
- Banking Service objects
 - Entity Objects
 - Account
 - Debit Card
 - Bank Transaction
 - Business Logic Objects
 - Withdrawal Transaction Manager
- Figs. 21.19-21.20

Copyright 2011 H. Gomma

dm-26

Steps in Using COMET/UML

- 1 Develop Software Requirements Model
- 2 **Develop Software Analysis Model**
 - Develop static model of problem domain (Chapter 7)
 - Structure system into objects (Chapter 8)
 - Develop statecharts for state dependent objects (Chapter 10)
 - **Develop object interaction diagrams for each use case (Chapter 9, 11)**
- 3 Develop Software Design Model



Copyright 2011 H. Goma

27