COMP 3700: Software Modeling and Design

(Application Analysis)

Topics

- Application Interaction Model
- Application Class Model
- Application State Model

Topics

- Steps for constructing an application interaction model
  - Determine the system boundary
  - Find actors
  - Find use cases
  - Find initial and final events
  - Prepare normal scenarios
  - Add variation and exception scenarios
  - Find external events
  - Organize actors and use cases

Determining the System Boundary

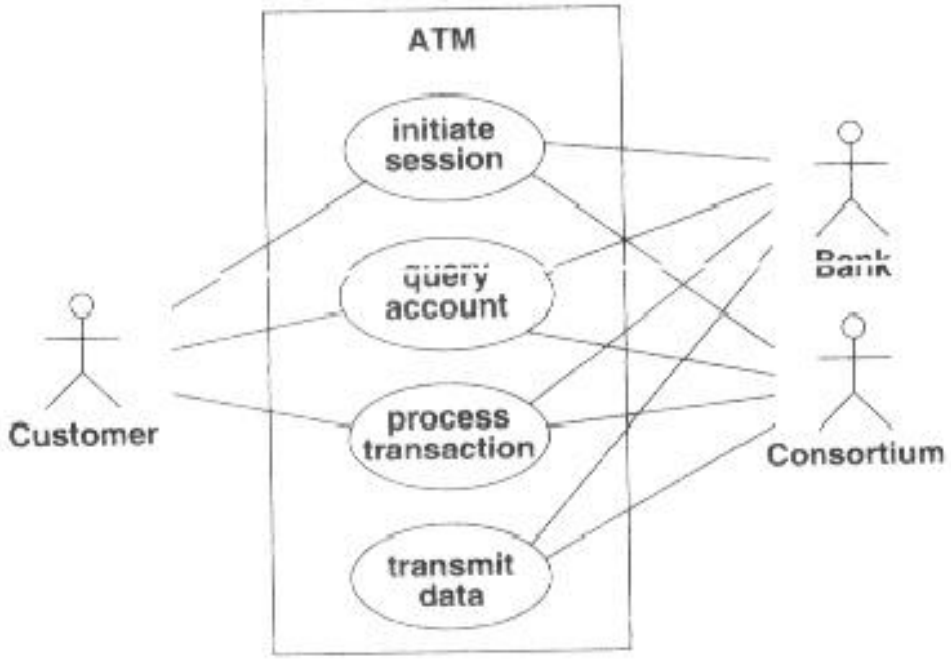
- You must know the precise scope of the application – the boundary of the system – to specify functionality.
- Humans are often actors that interact with the system.
- Determine the purpose of the system.
- ATM Example:
  - Concept statement indicates "design the software to support a computerized banking network including both human cashiers and automatic teller machines…."
  - Focus on ATM behavior and ignore cashier details.

Finding Actors

- Once the system boundary is identified, determine external objects that interact directly with the system.
- Actors include
  - Humans,
  - External devices, and
  - Software systems.
- ATM Example:
  - Customer
  - Bank
  - Consortium

Finding Use Cases

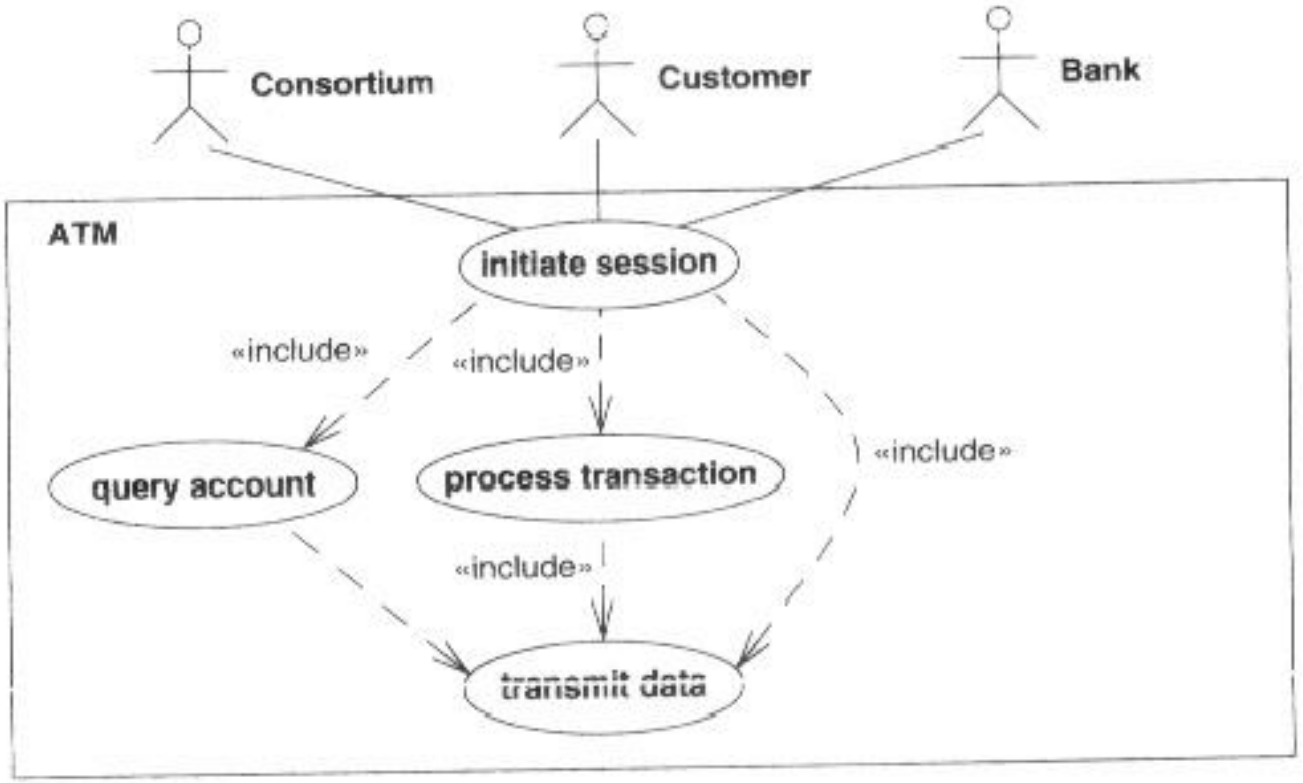
- **For each actor, list the fundamentally different ways in which the actor uses the system.**



Finding Use Cases

- **Initiate session:** The ATM establishes the identity of the user and makes available a list of accounts and actions.
- **Query account:** The system provides general data for an account, date of last transaction, and date of mailing.
- **Process transaction:** The ATM system performs an action that affects an account's balance, such as deposit, withdraw, and transfer.
- **Transmit data:** The ATM uses the consortium's facilities to communicate with the appropriate bank computers.

Organizing Use Cases



Finding Initial and Final Events

- **Initiate session:** The initial event is the insertion of a cash card. Two final events: system keeps the cash card or the system returns the cash card.
- **Query account:** The initial event is a customer's request for account data. The final event is the system's delivery of account data.
- **Process transaction:** The initial event is the customer's initiation transaction. Final event is commit or abort a transaction.
- **Transmit data:** Initial event is request for account data. Final event is successful transmission of data.

Preparing Normal Scenarios

Initiate session

The ATM asks the user to insert a card.
The user inserts a cash card.
The ATM accepts the card and reads its serial number.
The ATM requests the password.
The user enters "1234."
The ATM verifies the password by contacting the consortium and bank.
The ATM displays a menu of accounts and commands.
. . .
The user chooses the command to terminate the session.
The ATM prints a receipt, ejects the card, and asks the user to take them.
The user takes the receipt and the card.
The ATM asks the user to insert a card

Query account

The ATM displays a menu of accounts and commands.
The user chooses to query an account.
The ATM contacts the consortium and bank which return the data.
The ATM displays account data for the user.
The ATM displays a menu of accounts and commands.

Preparing Normal Scenarios

Process transaction

The ATM displays a menu of accounts and commands.
The user selects an account withdrawal.
The ATM asks for the amount of cash.
The user enters $100.
The ATM verifies that the withdrawal satisfies its policy limits.
The ATM contacts the consortium and bank and verifies that the account has sufficient funds.
The ATM dispenses the cash and asks the user to take it.
The user takes the cash.
The ATM displays a menu of accounts and commands.

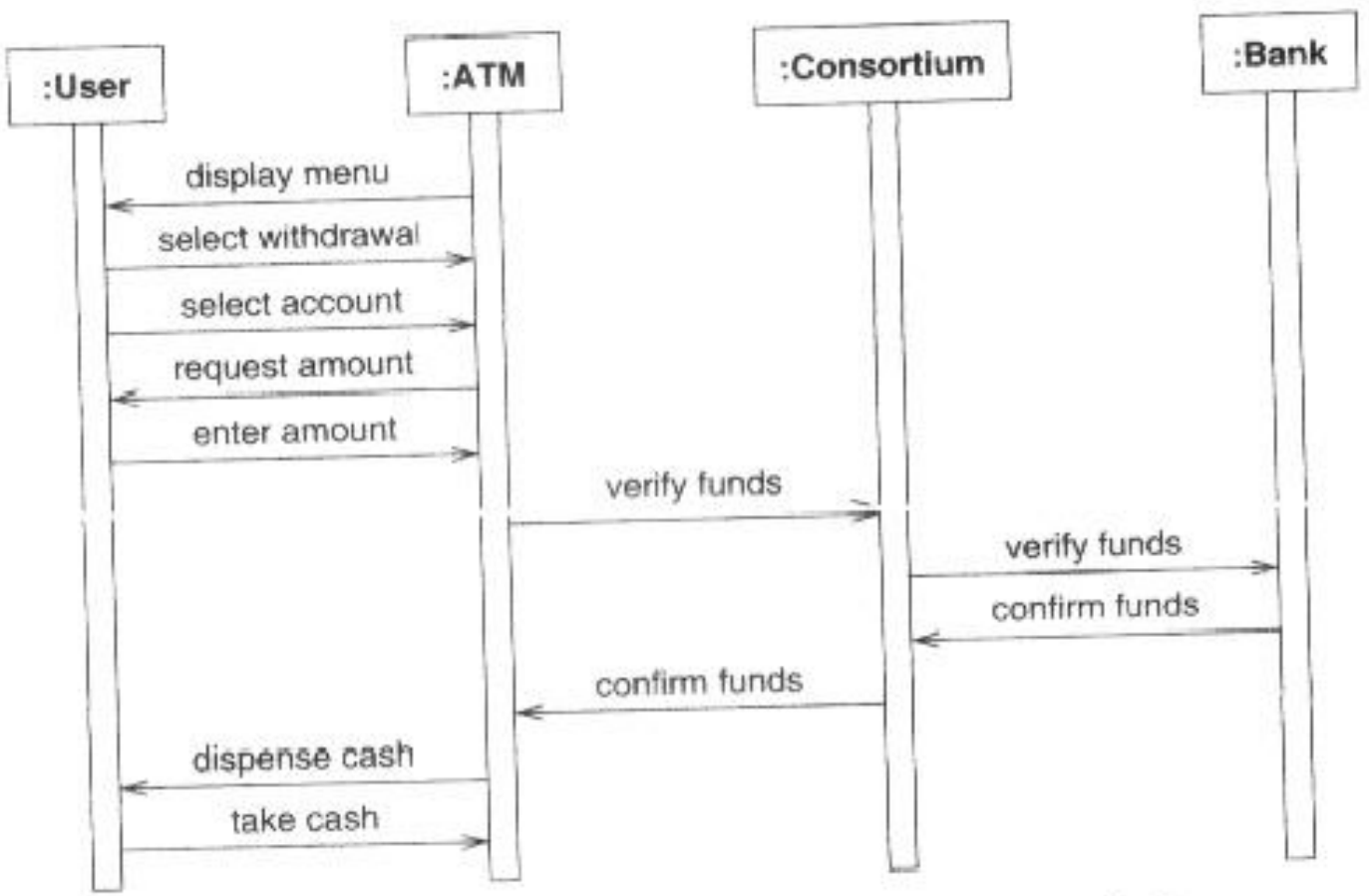
Transmit data

The ATM requests account data from the consortium.
The consortium accepts the request and forwards it to the appropriate bank.
The bank receives the request and retrieves the desired data.
The bank sends the data to the consortium.
The consortium routes the data to the ATM.

Adding Variations and Exception Scenarios

- Error is handling is most often the difficult part of developing software.
- ATM Example:
  - The ATM can't read the card.
  - The card has expired.
  - The ATM times out waiting for a response.
  - The amount is invalid.
  - The machine is out cash or paper.
  - The communication line is down.
  - The transaction is rejected.

Sequence Diagram for the Process Transaction Scenario

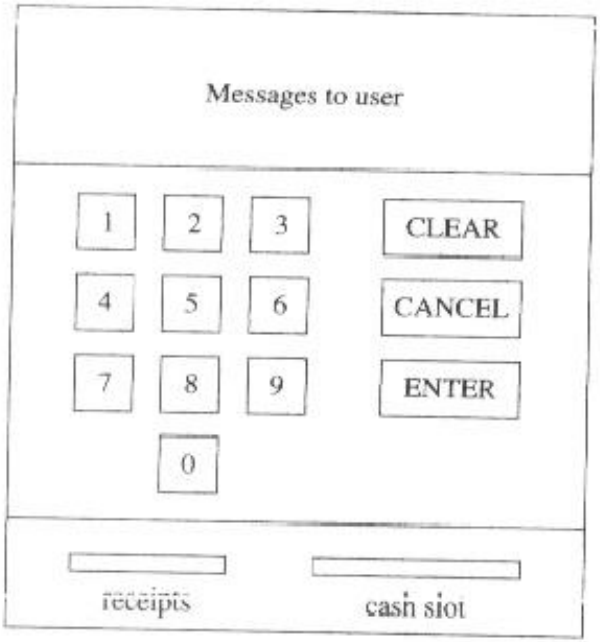


Application Class Model

- Application classes define the application itself, rather than the real-world objects that the application acts on.
- The steps for constructing an application class model:
  - Specify interface
  - Define boundary classes
  - Determine controllers
  - Check against the interaction model

Specify Interface

- Treat the interface at a coarse level of detail.
- Do not consider implementation or platform specific details.



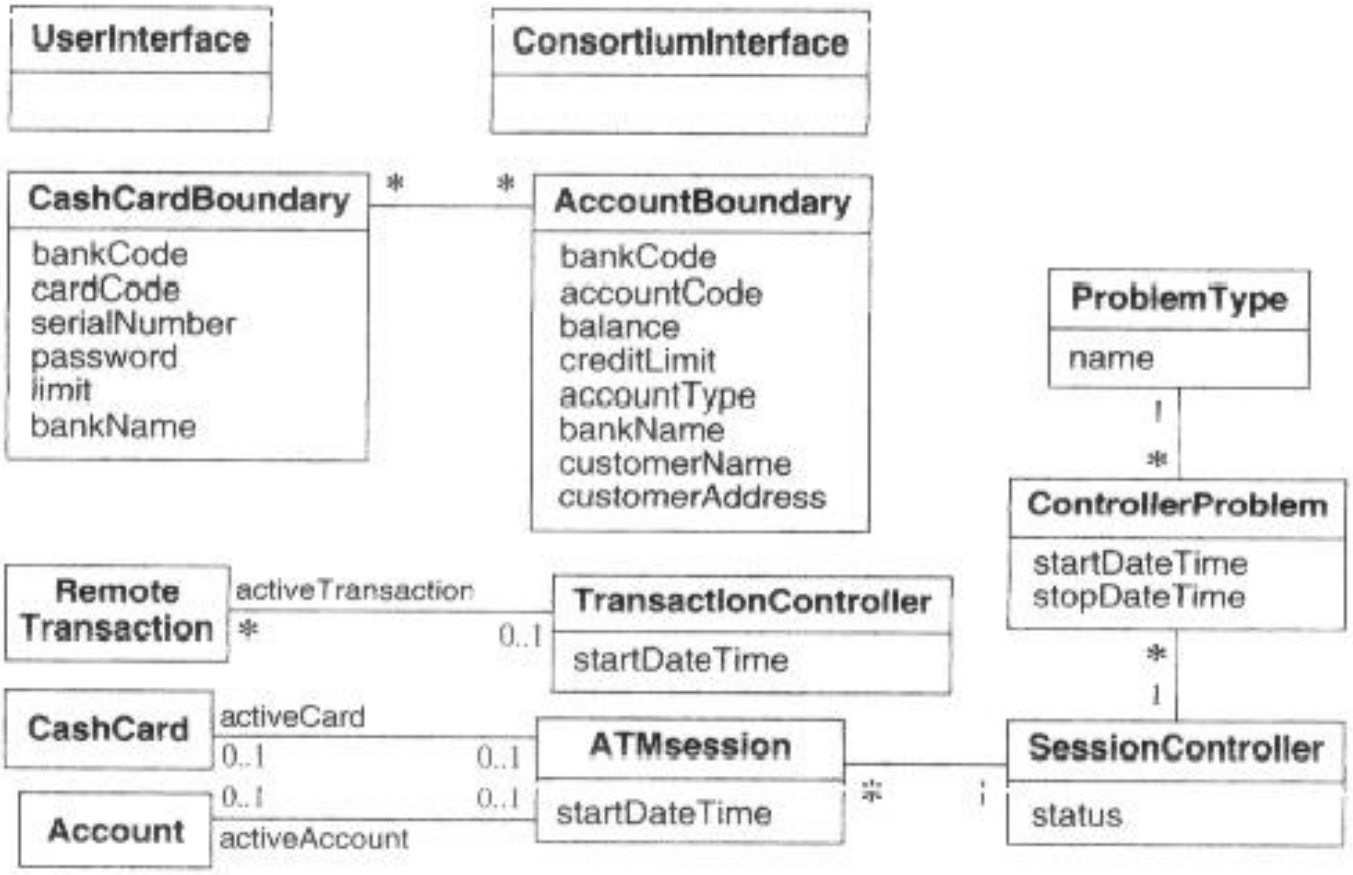
Identifying Boundary Objects

- Identify user interface controls that the user needs to initiate a use case (e.g., ReportEmergencyButton)
- Identify forms the user needs to enter data into the system (e.g., EmergencyReportForm)
- Identify notices and messages the system uses to respond to the user (e.g., AcknowledgementNotice)
- When multiple actors are involved in a use case ientify actor terminals to refer to the user interface under consideration.
- Do not model the visual aspects, use mock-ups.
- Always use end-user terms to describe boundary objects.

Identifying Controller Objects

- Control objects are responsible for coordinating domain (entity) and boundary objects.
- Often a close relation exists between a use case and a control object.
- Controller objects are often crated at the beginning of a use case and ceases to exist at its end.
- Heuristics:
  - Identify one control object per use case.
  - Identify one control object per actor in the use case.
  - The life span of a control object should cover the extent of the use case.

ATM Application Class Model



Heuristics for Detailed Sequence Diagrams

- The first column should correspond to the actor who initiated the use case.
- The second column should be a boundary object that the actor used to initiate the use case.
- The third column should be the control object that manages the rest of the use case.
- Control objects are created by boundary objects initiating the use case.
- Boundary objects are created by controller objects.
- Entity objects are accessed by control and boundary objects.
- The access from entity objects to boundary or control objects should be minimized.

FRIEND System Case Study

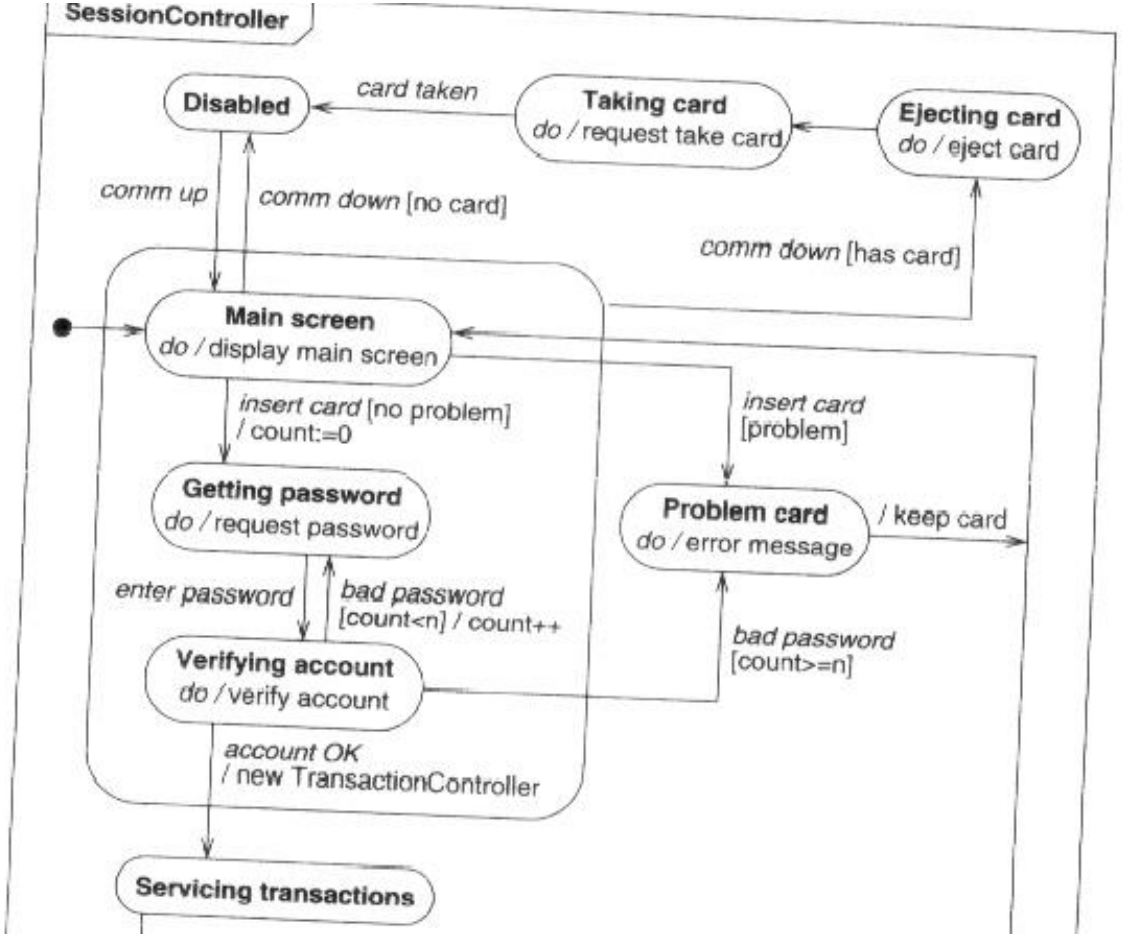
Application State Model

- The steps are as follows:
  - Determine the application classes with state.
  - Find events.
  - Build state diagrams.
  - Check against other state diagrams.
  - Check against the class model.
  - Check against the interaction model.

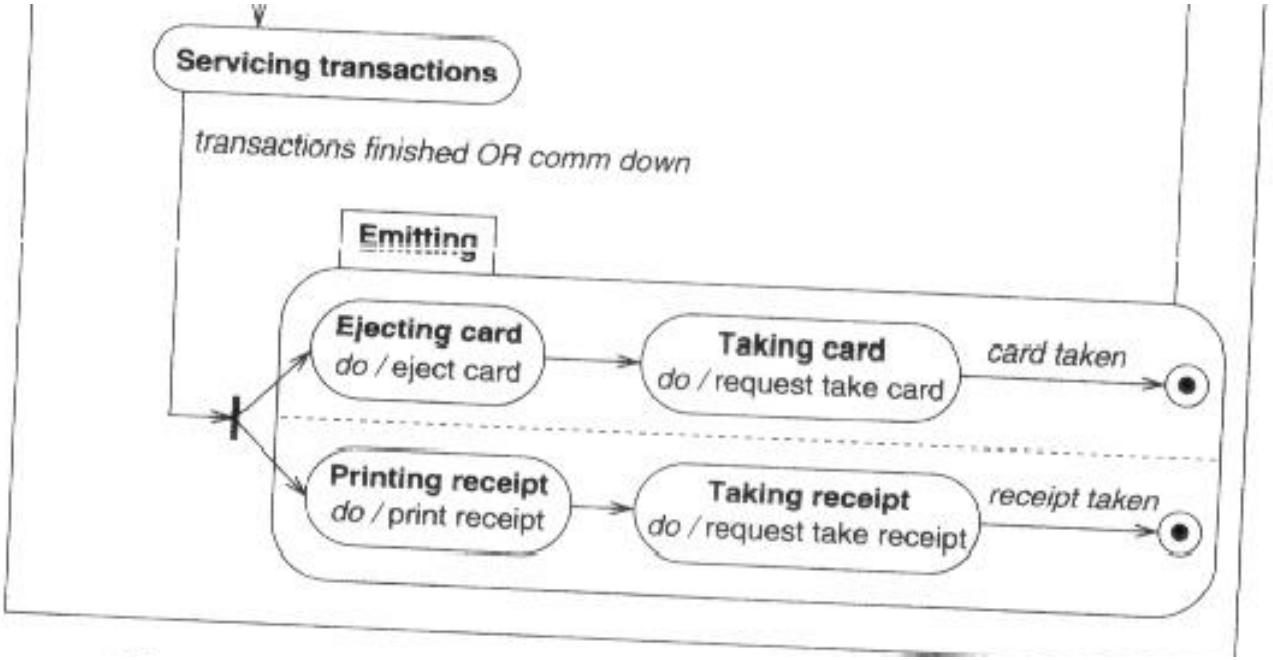
Application classes with State

- Interface (boundary) objects lack state dependent behavior.
- Controllers have important states
- Choose a class and consider a sequence diagram
- Arrange the events involving the class into a path whose arcs are labeled by events
- Find loops within the diagram.
- Merge other sequence diagrams, and find in each sequence diagram where it diverges from previous ones.
- Add variation or exceptional conditions

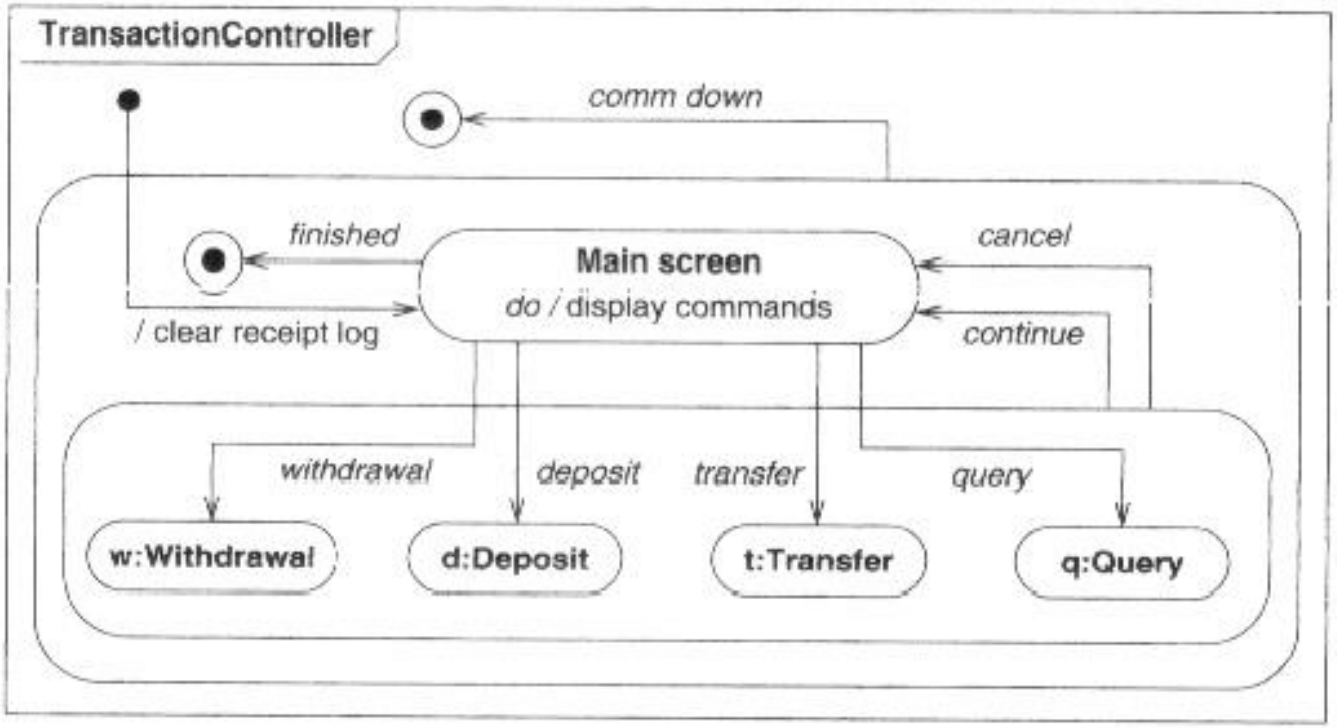
Application classes with State



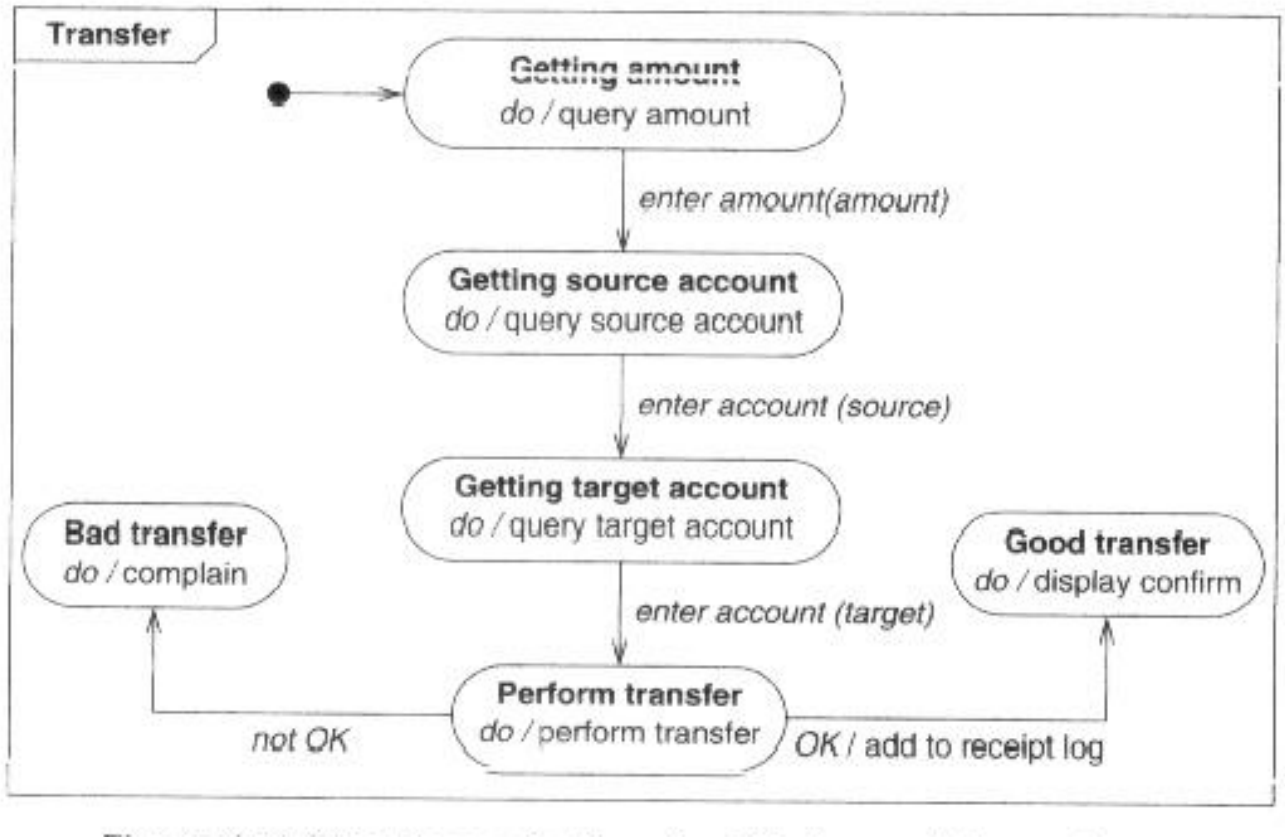
Application classes with State



Application classes with State



Application classes with State



Summary (Domain and Application Analysis)

- Purpose of analysis: Understand the problem so that correct design can be constructed.
- Good analysis: Captures the essential features of the problem domain without introducing implementation artifacts.
- Implementation artifacts prematurely limits design decisions.
- Two phases of analysis: Domain and Application
- Domain analysis involves class and state models, but it has seldom interaction models.

Summary (Domain and Application Analysis)

- Application analysis focuses on major application artifacts that are important, visible to users, and must be approved by them.
- The interaction model dominates application analysis.
- Application interaction model shows
  - the interaction between the system and outside world (High-level SSD)
  - Extension of high-level SSD with interactions among system objects (entity, boundary, controller).
- For application analysis
  - determine system boundary
  - define actors

Summary (Domain and Application Analysis)

- For application analysis (contn'd)
  - define use cases
  - For each use case, make up scenarios for normal cases, variations, extreme cases, and exceptions.
- Augment the domain classes with application classes
  - Application classes arise from boundary, interface, and controller classes.
  - Carefully check the use cases and scenarios to find them.
- Last phase of application analysis is to build an application state model.