Case Study

Lecture # 6a

Department of Computer Science and Technology
University of Bedfordshire

Written by David Goodwin,
based on the lectures of Marc Conrad and
on the book *Applying UML and Patterns (3rd ed.)*
by *C. Larman* (2005).

Modelling and Simulation, 2012

Case Study - ATM

- User Requirments
- Use Cases and interaction diagrams
- Analysis Classes
- Class Diagram
- State-Machine Diagrams
- Detailed Design

Case Study - ATM

User Requirments I

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a customer console (keyboard and display) for interaction with the customer, a slot for depositing envelopes, a dispenser for cash (in multiples of $20), a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link. (The software on the latter is not part of the requirements for this problem.)

User Requirments II

The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned - except as noted below.

User Requirments III

The ATM must be able to provide the following services to the customer:

- A customer must be able to make a cash withdrawal from any suitable account linked to the card, in multiples of $20.00. Approval must be obtained from the bank before cash is dispensed.
- A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or checks in an envelope. The customer will enter the amount of the deposit into the ATM, subject to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.
- A customer must be able to make a transfer of money between any two accounts linked to the card.
- A customer must be able to make a balance inquiry of any account linked to the card.

A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine.

User Requirments IV

The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once it has been approved. In the case of a deposit, a second message will be sent to the bank indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the timeout period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.)

User Requirments V

If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back.

User Requirments VI

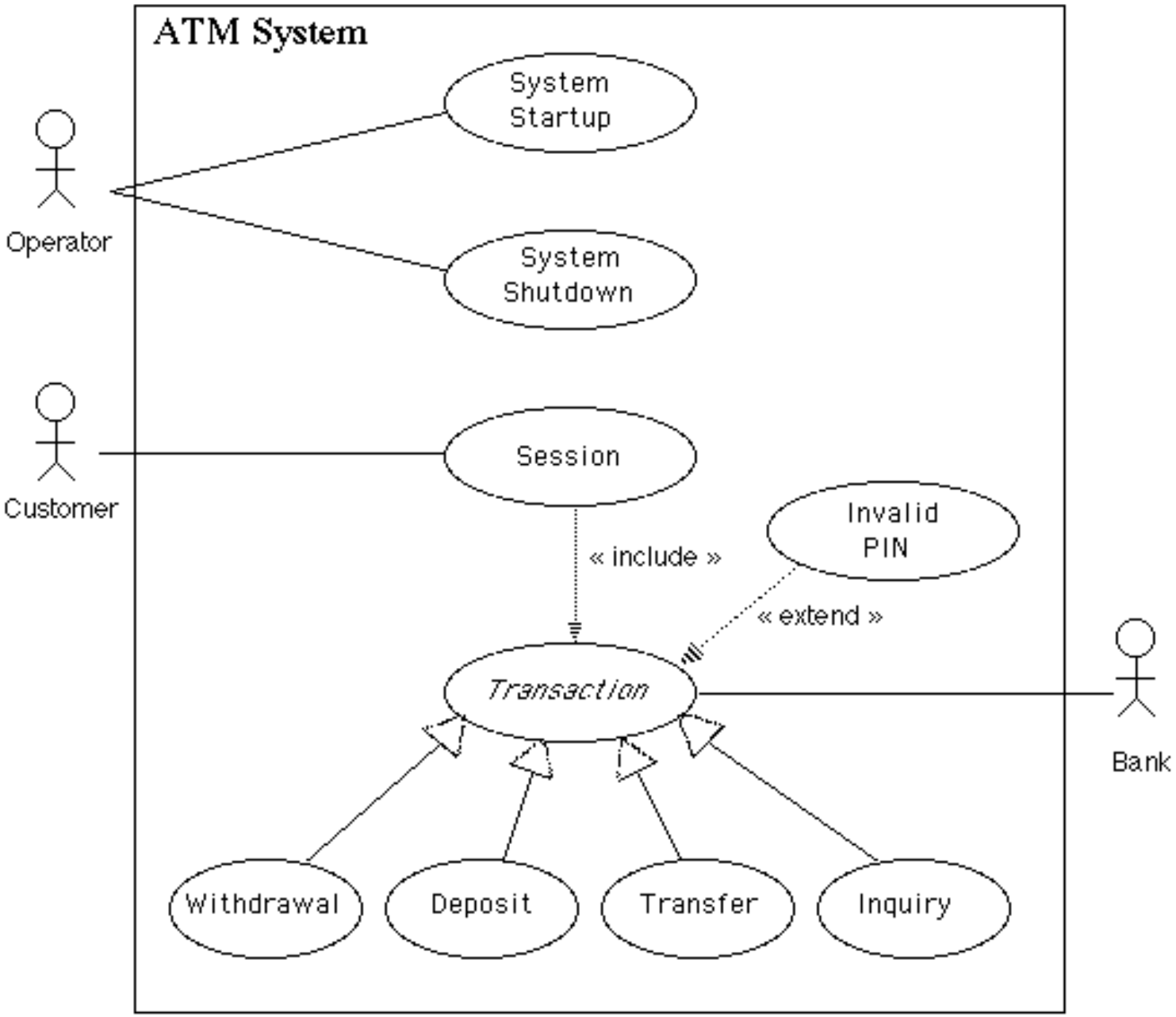
If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction.
The ATM will provide the customer with a printed receipt for each successful transaction, showing the date, time, machine location, type of transaction, account(s), amount, and ending and available balance(s) of the affected account ("to" account for transfers).

User Requirments VII

The ATM will have a key-operated switch that will allow an operator to start and stop the servicing of customers. After turning the switch to the "on" position, the operator will be required to verify and enter the total cash on hand. The machine can only be turned off when it is not servicing a customer. When the switch is moved to the "off" position, the machine will shut down, so that the operator may remove deposit envelopes and reload the machine with cash, blank receipts, etc.

User Requirments IX

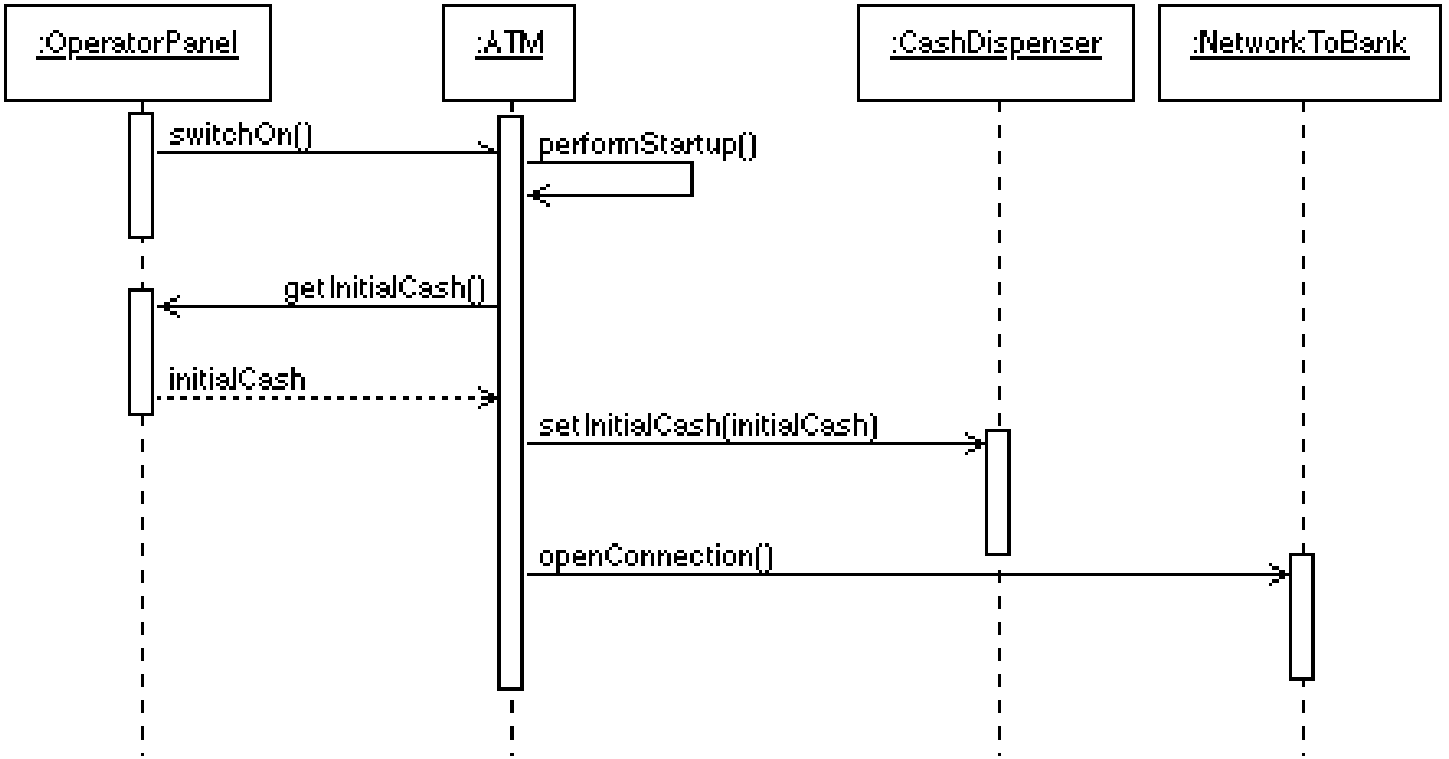
The ATM will also maintain an internal log of transactions to facilitate resolving ambiguities arising from a hardware failure in the middle of a transaction. Entries will be made in the log when the ATM is started up and shut down, for each message sent to the Bank (along with the response back, if one is expected), for the dispensing of cash, and for the receiving of an envelope. Log entries may contain card numbers and dollar amounts, but for security will never contain a PIN.



System Startup Use Case

- The system is started up when the operator turns the operator switch to the “on” position. The operator will be asked to enter the amount of money currently in the cash dispenser, and a connection to the bank will be established. Then the servicing of customers can begin.

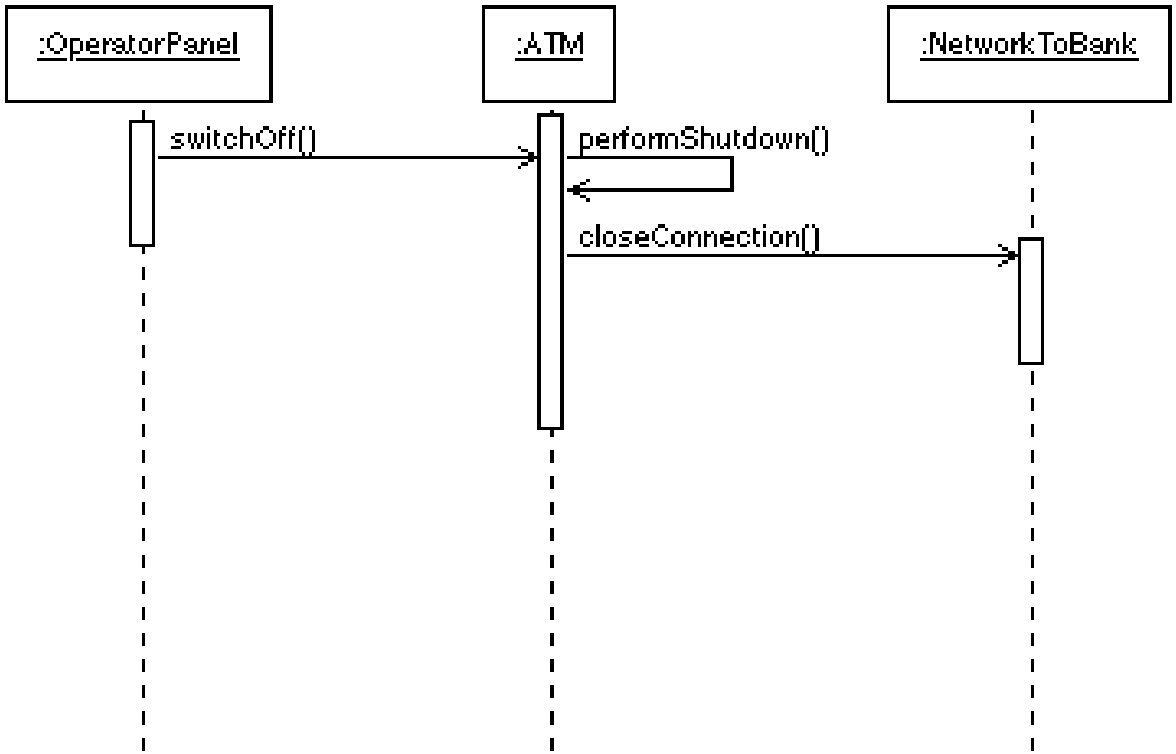
System Startup Sequence Diagram



System Shutdown Use Case

- The system is shut down when the operator makes sure that no customer is using the machine, and then turns the operator switch to the "off" position. The connection to the bank will be shut down. Then the operator is free to remove deposited envelopes, replenish cash and paper, etc.

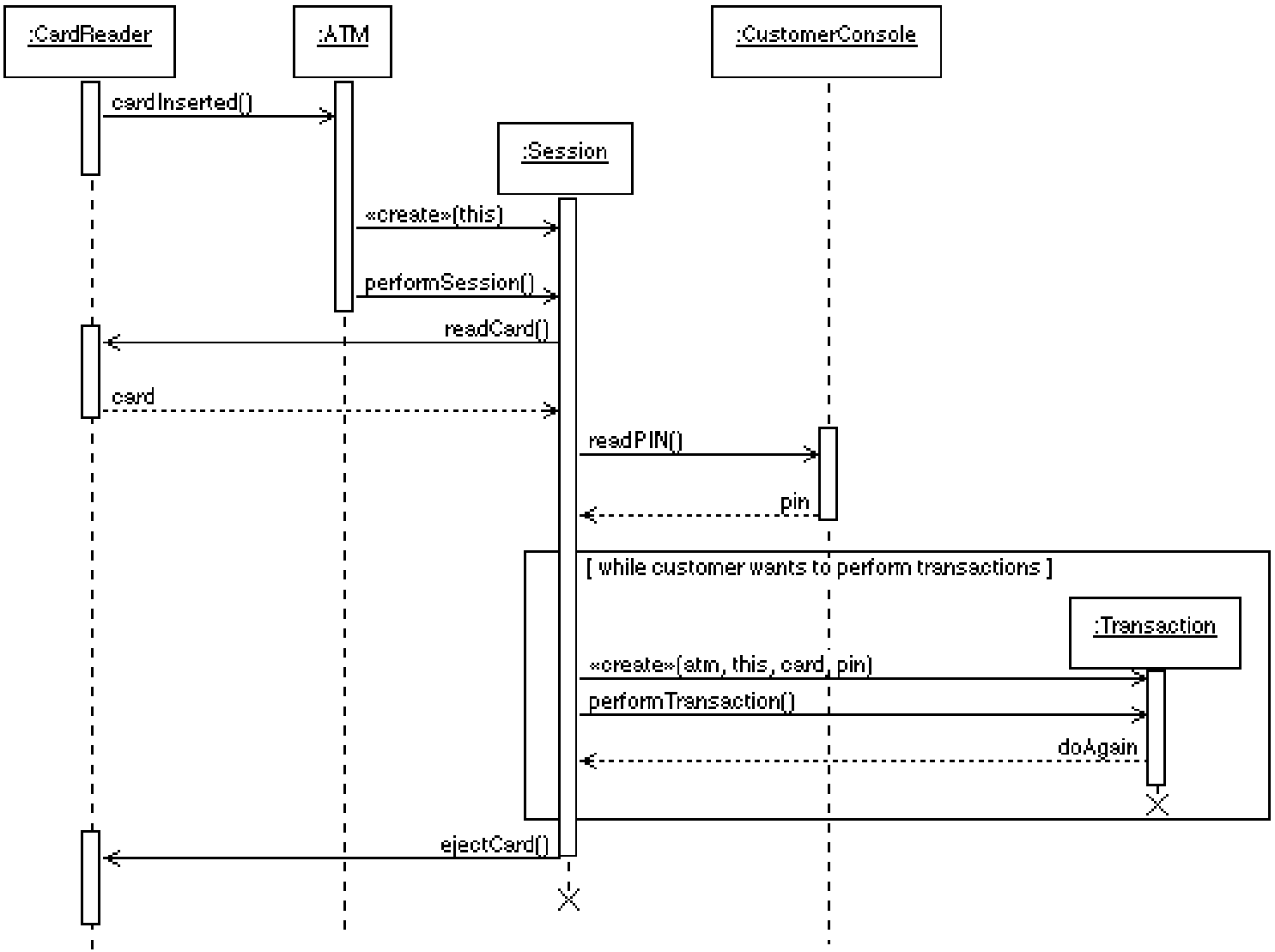
System Shutdown Sequence Diagram



Session Use Case

- A session is started when a customer inserts an ATM card into the card reader slot of the machine. The ATM pulls the card into the machine and reads it. (If the reader cannot read the card due to improper insertion or a damaged stripe, the card is ejected, an error screen is displayed, and the session is aborted.) The customer is asked to enter his/her PIN, and is then allowed to perform one or more transactions, choosing from a menu of possible types of transaction in each case. After each transaction, the customer is asked whether he/she would like to perform another. When the customer is through performing transactions, the card is ejected from the machine and the session ends. If a transaction is aborted due to too many invalid PIN entries, the session is also aborted, with the card being retained in the machine.
- The customer may abort the session by pressing the Cancel key when entering a PIN or choosing a transaction type.

Session Sequence Diagram

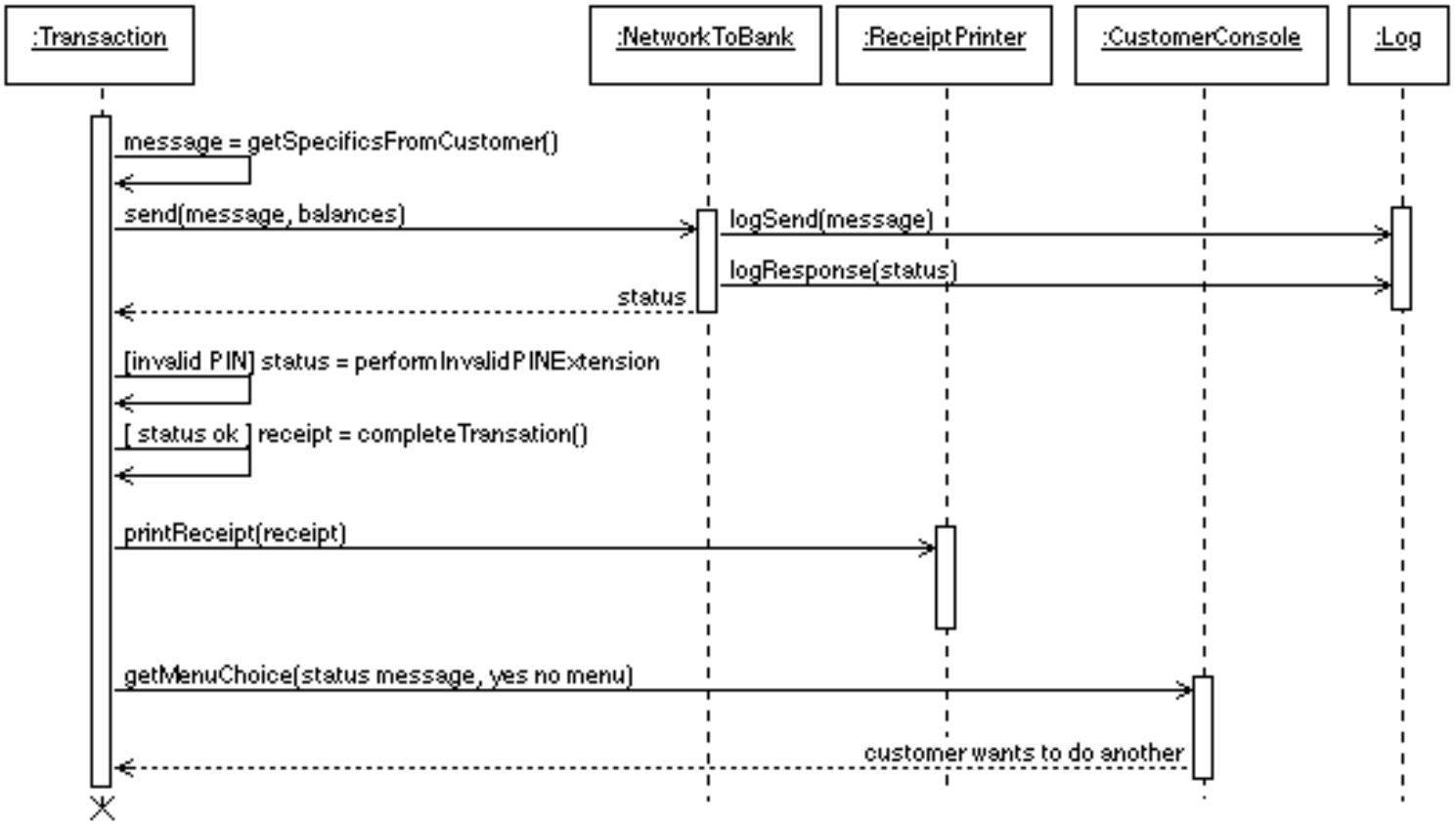


Transaction Use Case

- *Note: Transaction is an abstract generalization. The flow of events given here describes the behavior common to all types of transaction. The flows of events for the individual types of transaction give the features that are specific to that type of transaction.*
- A transaction use case is started within a session when the customer chooses a transaction type from a menu of options. The customer will be asked to furnish appropriate details. The transaction will be sent to the bank, with information from the customer's card and PIN entered.
- If the bank approves the transaction, any steps needed to complete the transaction will be performed, and then a receipt will be printed. Then the customer will be asked whether he/she wishes to do another transaction.
- If the bank reports that the customer's PIN is invalid, the Invalid PIN extension will be performed and then an attempt will be made to continue the transaction. If the customer's card is retained due to too many invalid PINs, the transaction will be aborted, and the customer will not be offered the option of doing another.
- If a transaction is cancelled by the customer, or fails for any reason other than repeated entries of an invalid PIN, a screen will be displayed informing the customer of the reason for the failure of the transaction, and then the customer will be offered the opportunity to do another.
- The customer may cancel a transaction by pressing the Cancel key as described for each individual type of transaction below.
- All messages to the bank and responses back are recorded in the ATM's log.

University of Bedfordshire

Transaction Sequence Diagram

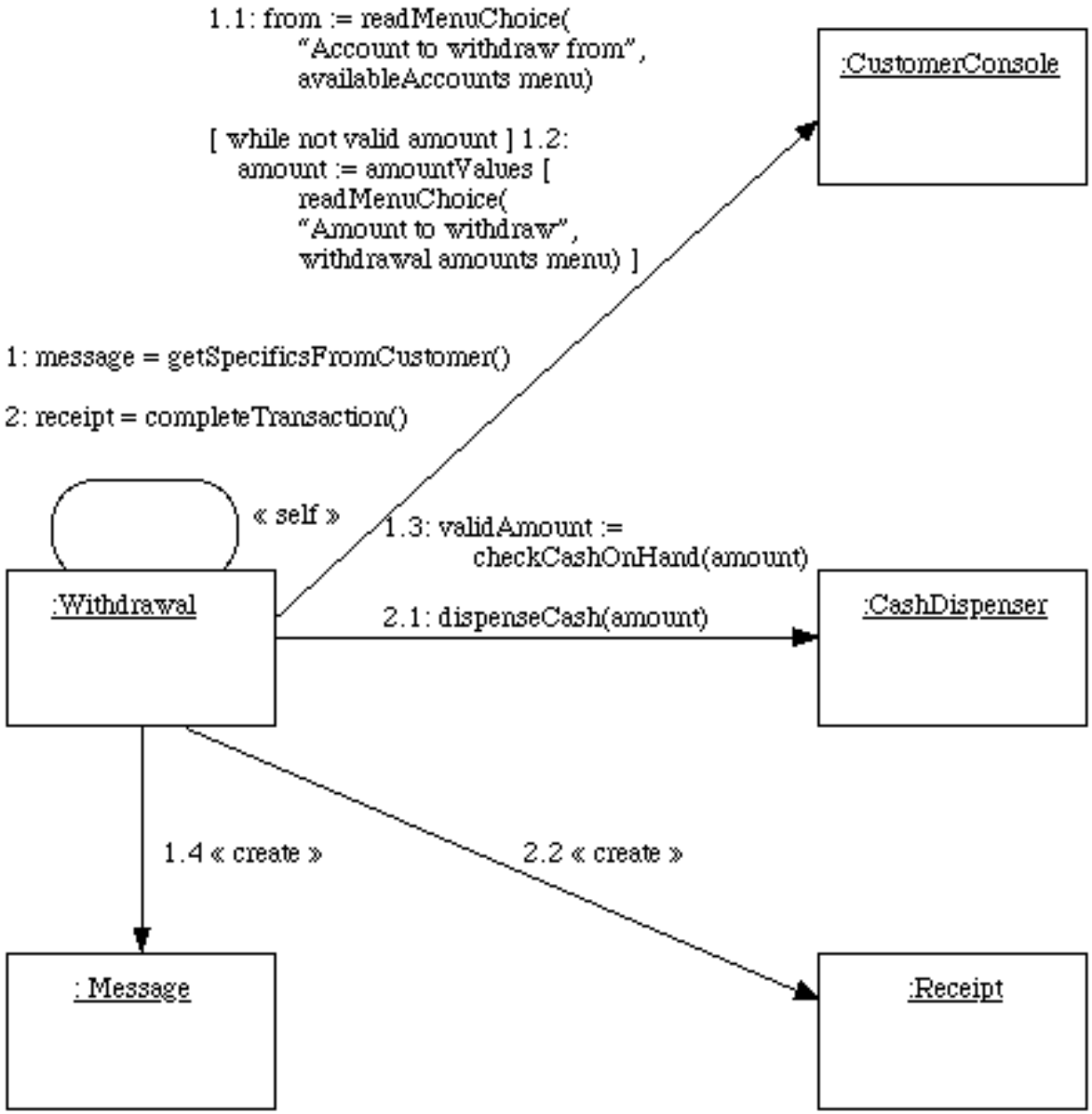


Withdrawal Transaction Use Case

- A withdrawal transaction asks the customer to choose a type of account to withdraw from (e.g. checking) from a menu of possible accounts, and to choose a dollar amount from a menu of possible amounts. The system verifies that it has sufficient money on hand to satisfy the request before sending the transaction to the bank. (If not, the customer is informed and asked to enter a different amount.) If the transaction is approved by the bank, the appropriate amount of cash is dispensed by the machine before it issues a receipt. (The dispensing of cash is also recorded in the ATM's log.)
- A withdrawal transaction can be cancelled by the customer pressing the Cancel key any time prior to choosing the dollar amount.

University of Bedfordshire

Withdrawal Transaction Collaboration

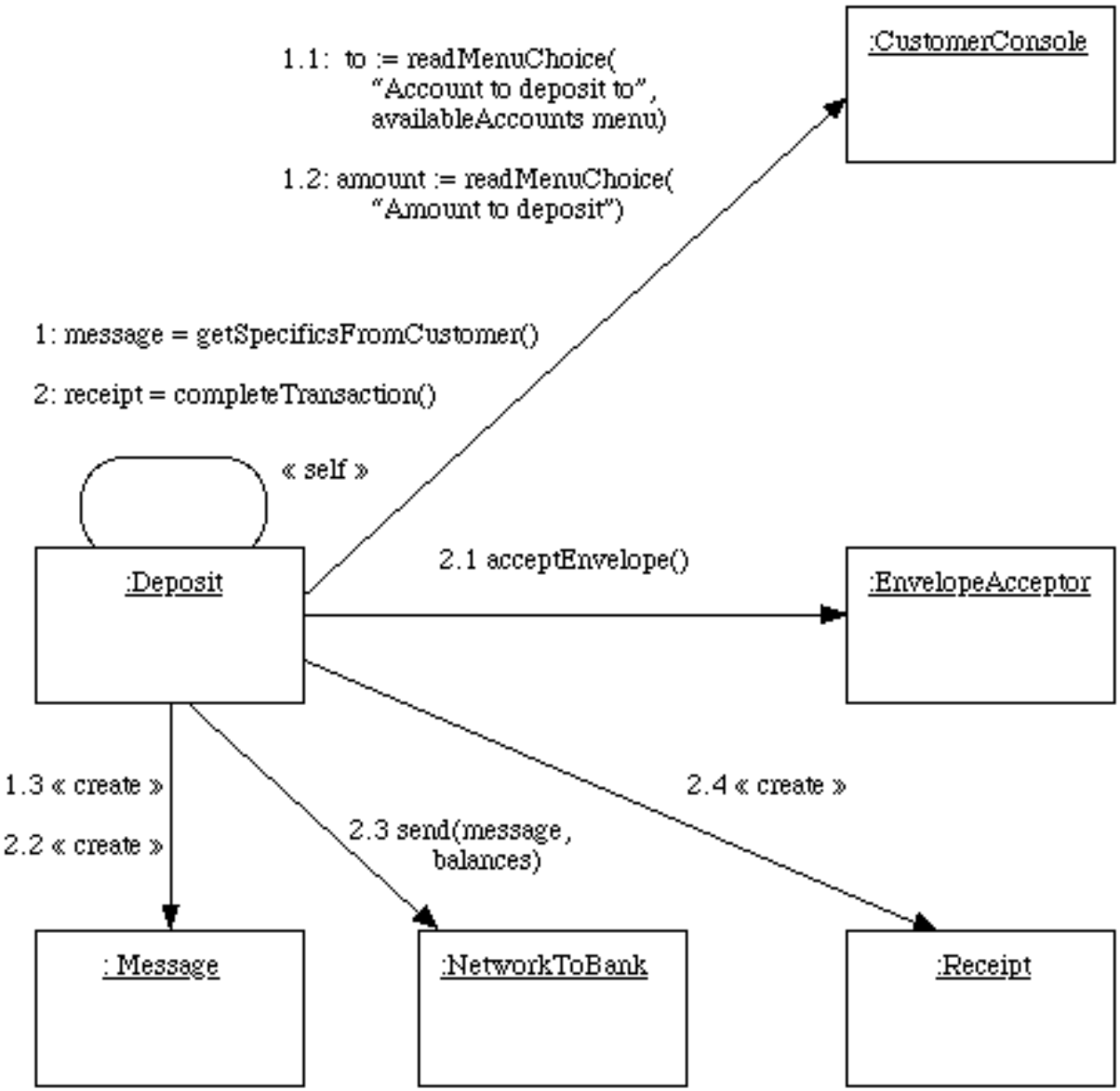


Deposit Transaction Use Case

- A deposit transaction asks the customer to choose a type of account to deposit to (e.g. checking) from a menu of possible accounts, and to type in a dollar amount on the keyboard. The transaction is initially sent to the bank to verify that the ATM can accept a deposit from this customer to this account. If the transaction is approved, the machine accepts an envelope from the customer containing cash and/or checks before it issues a receipt. Once the envelope has been received, a second message is sent to the bank, to confirm that the bank can credit the customer's account - contingent on manual verification of the deposit envelope contents by an operator later. (The receipt of an envelope is also recorded in the ATM's log.)
- A deposit transaction can be cancelled by the customer pressing the Cancel key any time prior to inserting the envelope containing the deposit. The transaction is automatically cancelled if the customer fails to insert the envelope containing the deposit within a reasonable period of time after being asked to do so.

University of Bedfordshire

Deposit Transaction Collaboration



Transfer Transaction Use Case

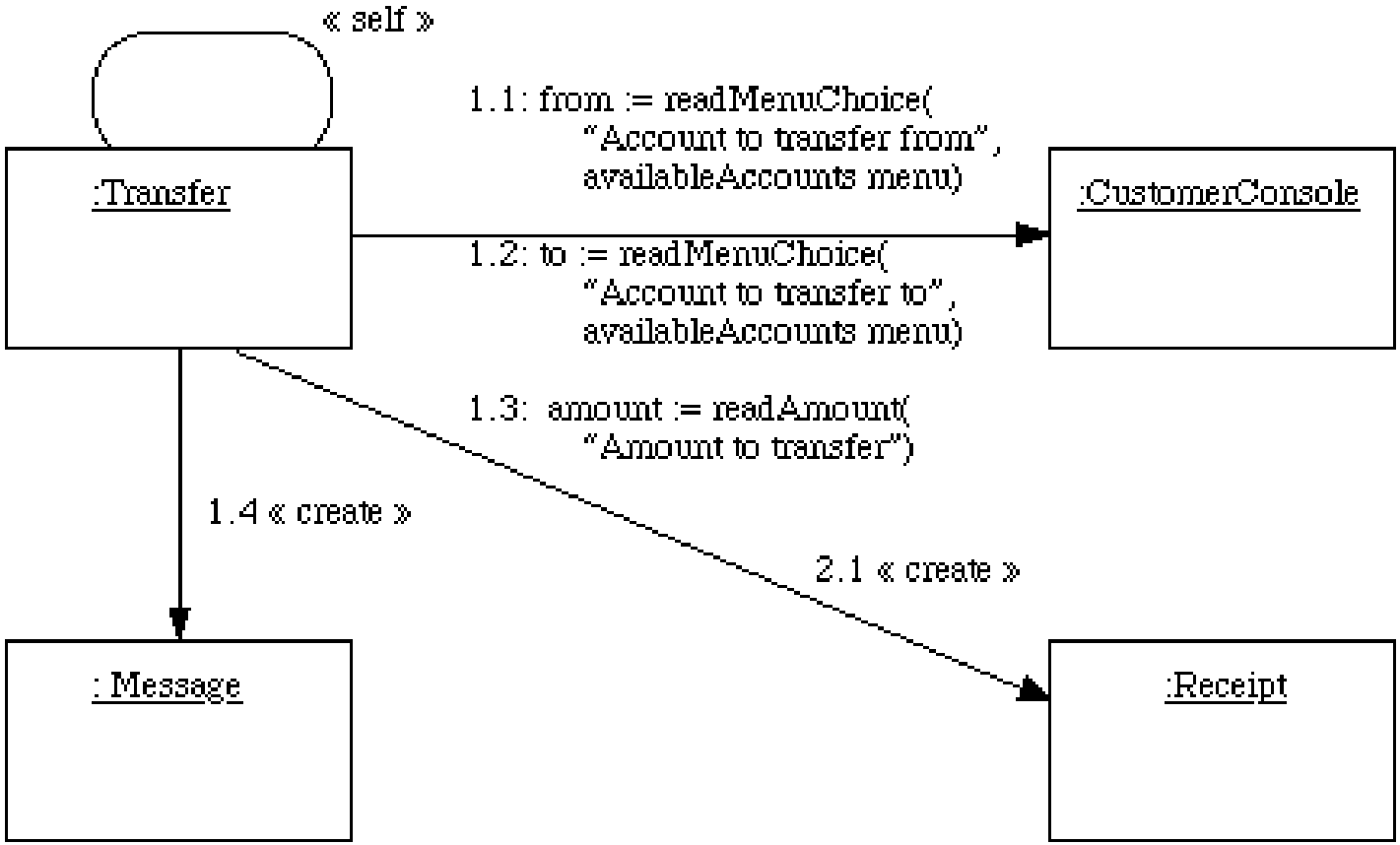
- A transfer transaction asks the customer to choose a type of account to transfer from (e.g. checking) from a menu of possible accounts, to choose a different account to transfer to, and to type in a dollar amount on the keyboard. No further action is required once the transaction is approved by the bank before printing the receipt.
- A transfer transaction can be cancelled by the customer pressing the Cancel key any time prior to entering a dollar amount.

University of Bedfordshire

Transfer Transaction Collaboration

1: message = getSpecificsFromCustomer()

2: receipt = completeTransaction()



Inquiry Transaction Use Case

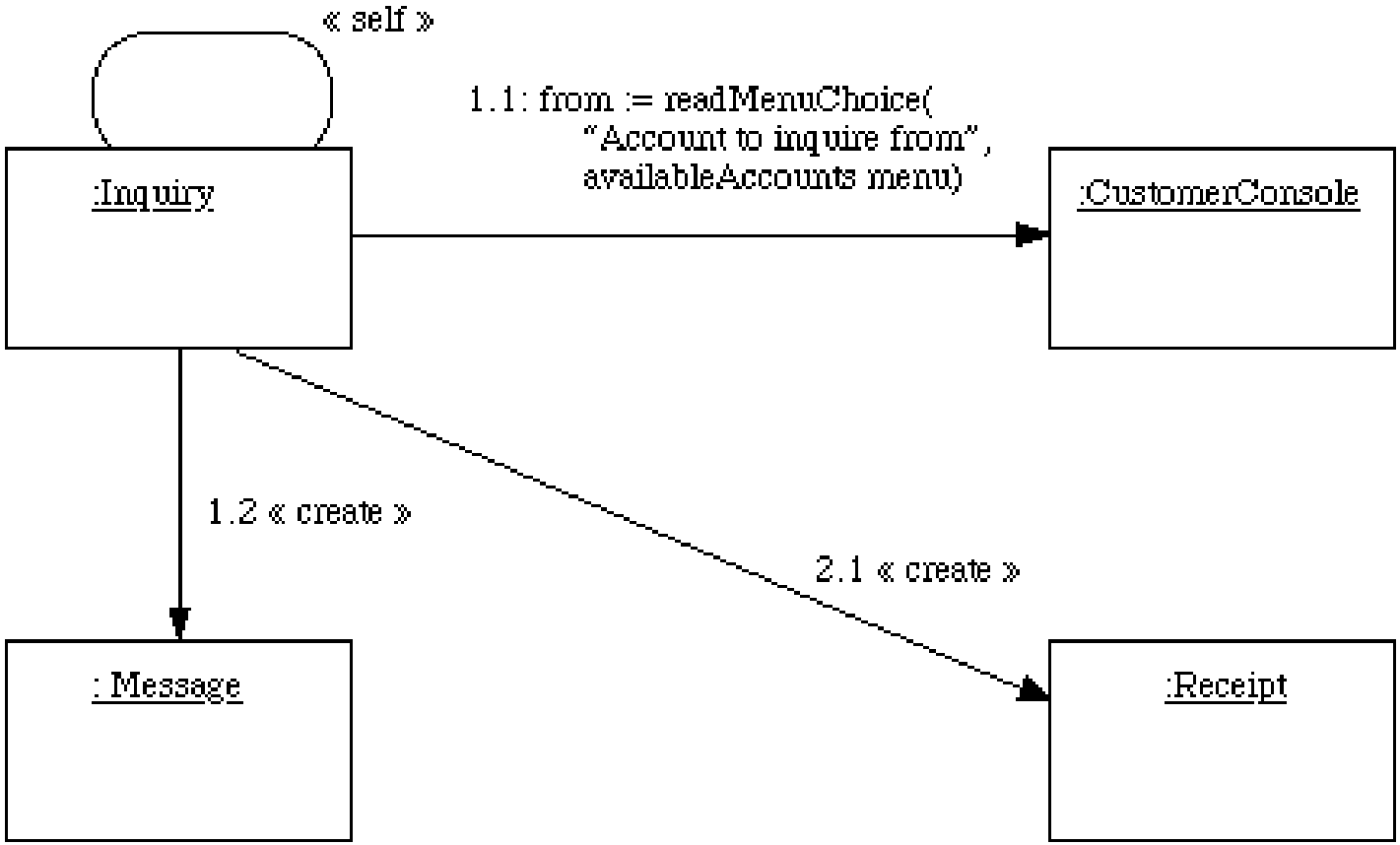
- An inquiry transaction asks the customer to choose a type of account to inquire about from a menu of possible accounts. No further action is required once the transaction is approved by the bank before printing the receipt.
- An inquiry transaction can be cancelled by the customer pressing the Cancel key any time prior to choosing the account to inquire about.

University of Bedfordshire

Inquiry Transaction Collaboration

1: message = getSpecificsFromCustomer()

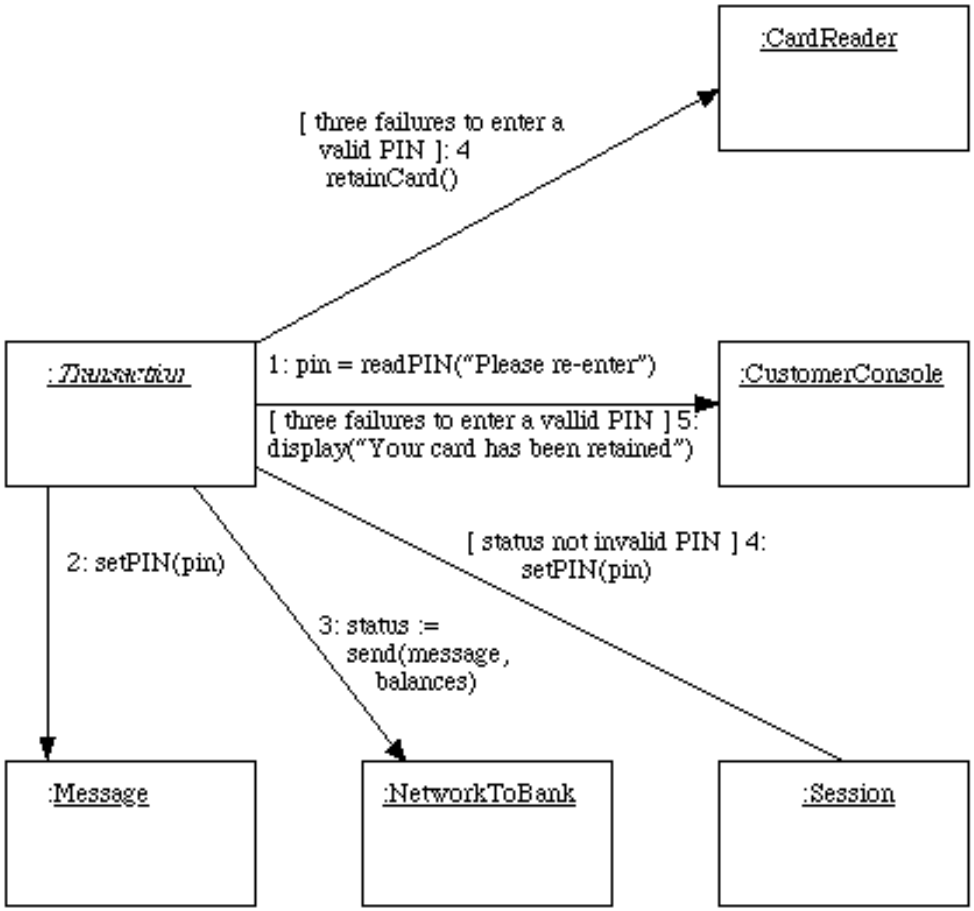
2: receipt = completeTransaction()



Invalid PIN Extension

- An invalid PIN extension is started from within a transaction when the bank reports that the customer's transaction is disapproved due to an invalid PIN. The customer is required to re-enter the PIN and the original request is sent to the bank again. If the bank now approves the transaction, or disapproves it for some other reason, the original use case is continued; otherwise the process of re-entering the PIN is repeated. Once the PIN is successfully re-entered, it is used for both the current transaction and all subsequent transactions in the session. If the customer fails three times to enter the correct PIN, the card is permanently retained, a screen is displayed informing the customer of this and suggesting he/she contact the bank, and the entire customer session is aborted.
- If the customer presses Cancel instead of re-entering a PIN, the original transaction is cancelled.

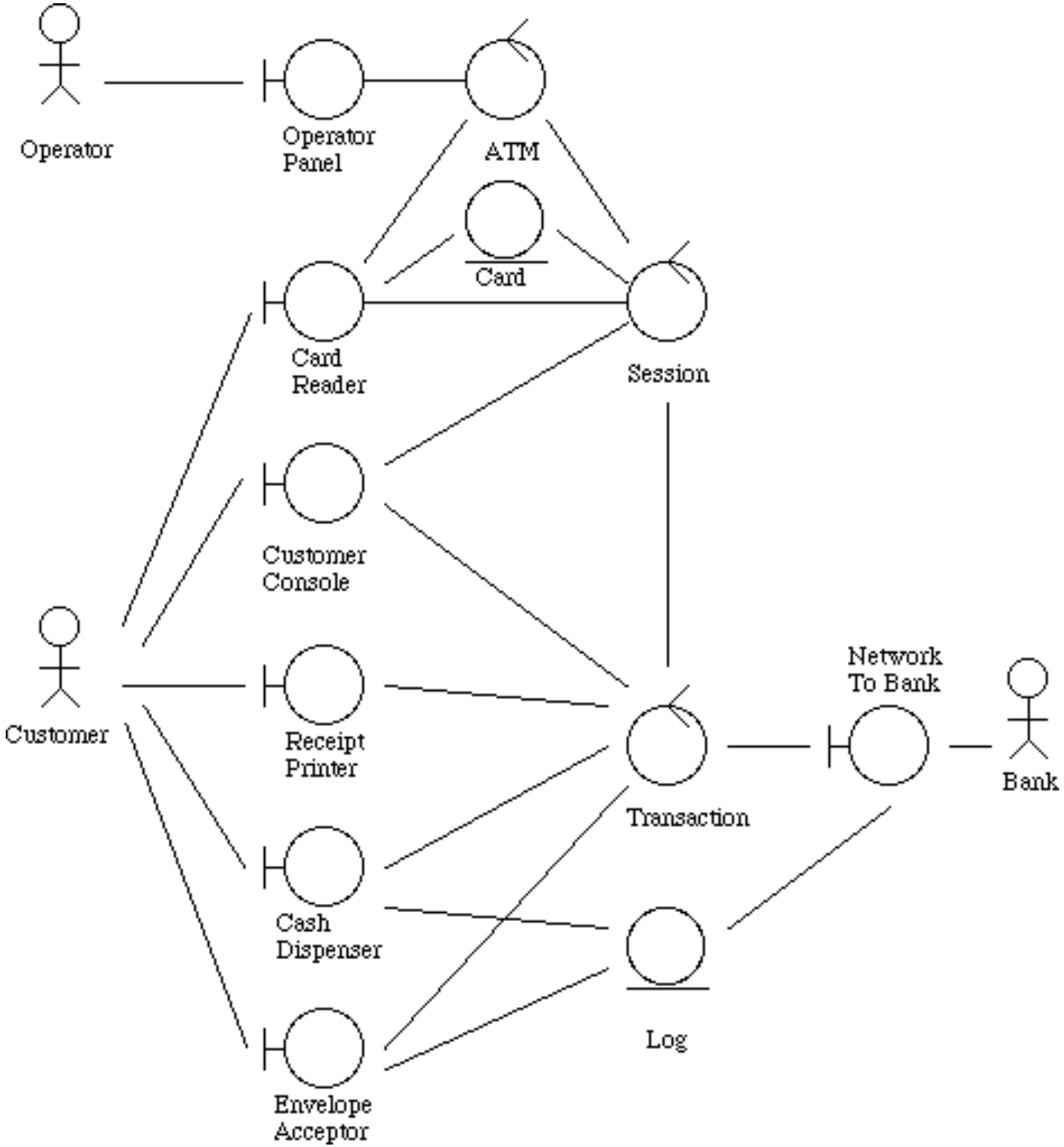
Invalid PIN Extension Collaboration



As soon as the customer enters a valid PIN (send() returns a status other than incorrect PIN, the extension is terminated. If the customer re-enters invalid PINs three times, the ATM card is retained and the extension (and session of which it is a part) is aborted. If the user presses Cancel, this extension is aborted immediately, and the transaction that initiated is aborted immediately as well.

Analysis Classes

- An initial reading of the use cases suggests that the following will be part of the system.
  - A controller object representing the ATM itself (managing the boundary objects listed below.)
  - Boundary objects representing the individual component parts of the ATM:
    - Operator panel.
    - Card reader.
    - Customer console, consisting of a display and keyboard.
    - Network connection to the bank.
    - Cash dispenser.
    - Envelope acceptor.
    - Receipt printer.
  - Controller objects corresponding to use cases. (Note: class ATM can handle the Startup and Shutdown use cases itself, so these do not give rise to separate objects here.)
    - Session
    - Transaction (abstract generalization, responsible for common features, with concrete specializations responsible for type-specific portions)
  - An entity object representing the information encoded on the ATM card inserted by customer.
  - An entity object representing the log of transactions maintained by the machine.

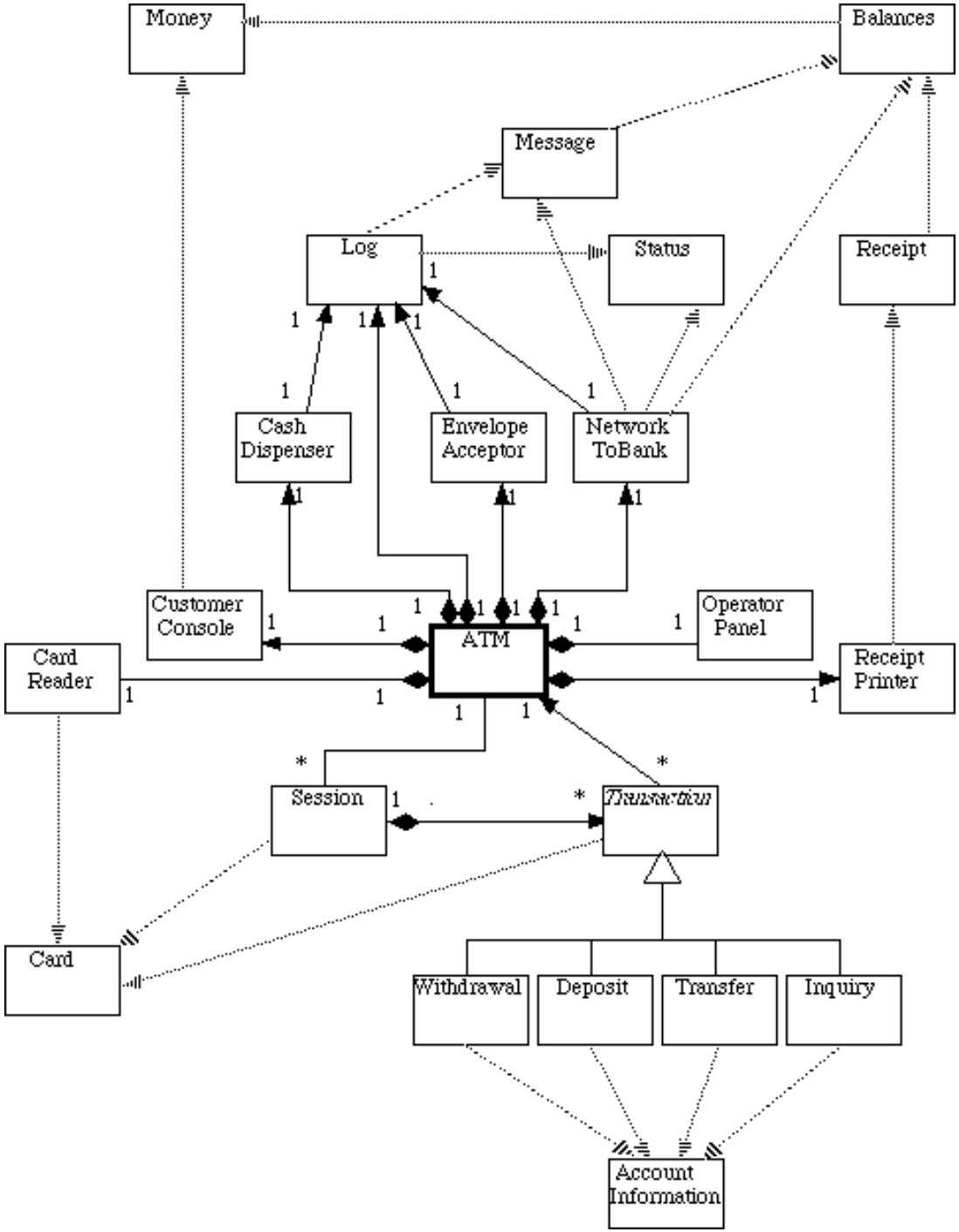

Operator
Operator Panel
ATM
Card
Card Reader
Session
Customer Console
Customer
Receipt Printer
Network To Bank
Bank
Transaction
Cash Dispenser
Log
Envelope Acceptor

Class Diagram for Example ATM System

- Shown below is the class diagram for the ATM system. The basic structure of the class diagram arises from the responsibilities and relationships discovered when doing the CRC cards and Interaction Diagrams. (If a class uses another class as a collaborator, or sends a message to an object of that class during an Interaction, then there must either be an association linking objects of those classes, or linking the "sending" class to an object which provides access to an object of the "receiving" class.)
- In the case of the ATM system, one of the responsibilities of the ATM is to provide access to its component parts for Session and Transaction objects; thus, Session and Transaction have associations to ATM, which in turn has associations to the classes representing the individual component parts. (Explicit "uses" links between Session and Transaction, on the one hand, and the component parts of the ATM, on the other hand, have been omitted from the diagram to avoid making it excessively cluttered.)

Class Diagram for Example ATM System

- Some classes were discovered when doing analysis (see the Analysis Class Diagram developed earlier.)
- Some classes were discovered when doing CRC cards
  - Message - used to represent a message to the bank.
  - Receipt - used to encapsulate information to be printed on a receipt.
  - Status - used to represent return value from message to the bank.
  - Balances - used to record balance information returned by the bank.
- Some classes were discovered when doing detailed design or writing code
  - Money - used to represent money amounts, in numerous places.
  - AccountInformation - contains names of various types of accounts customer can choose from
- That is, OO design is not a "waterfall" process - discoveries made when doing detailed design and coding can impact overall system design.
- To prevent the diagram from becoming overly large, only the name of each class is shown - the attribute and behavior "compartments" are shown in the detailed design, but are omitted here.

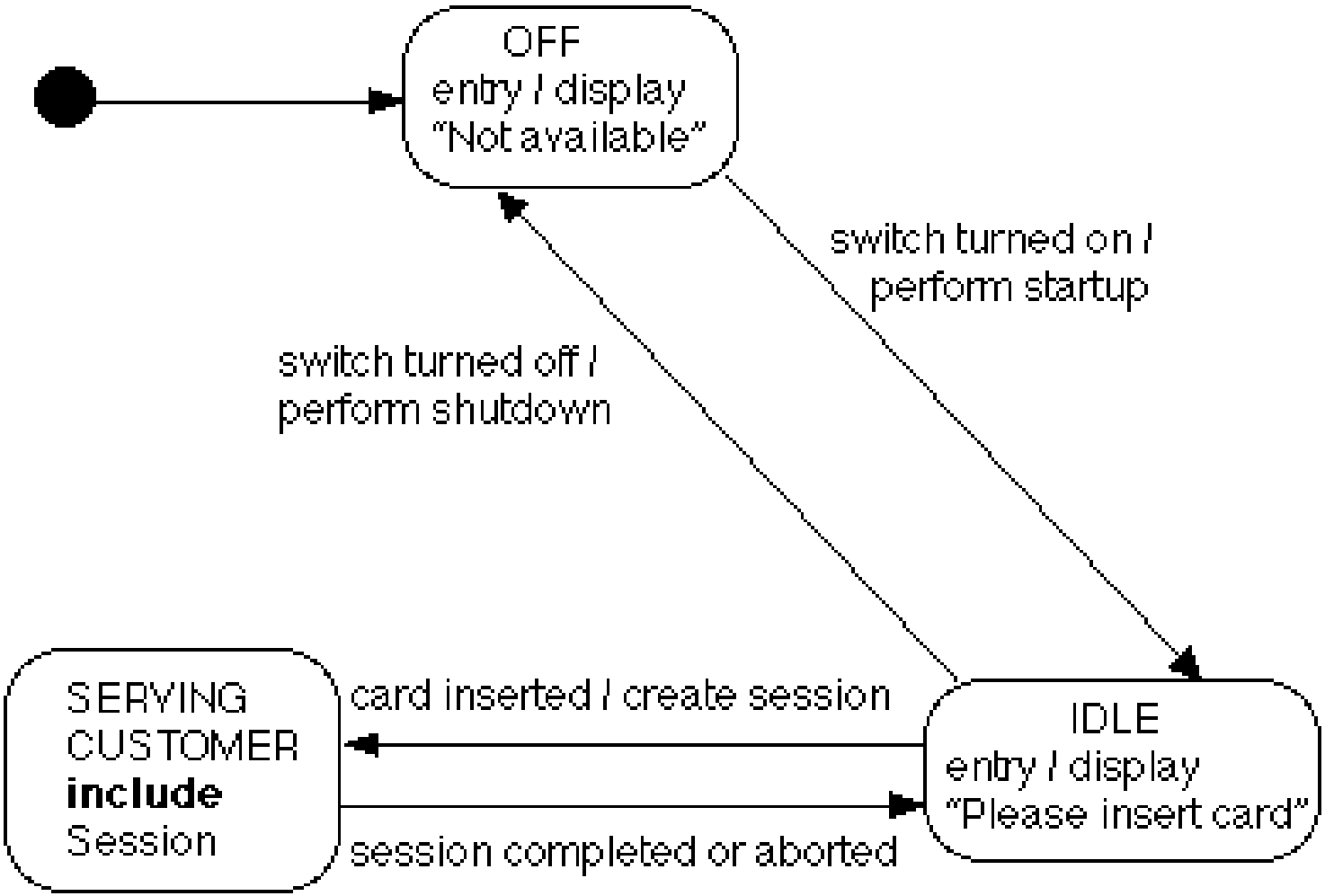

Money
Balances
Message
Log
Status
Receipt
Cash Dispenser
Envelope Acceptor
Network ToBank
Customer Console
ATM
Operator Panel
Card Reader
Receipt Printer
Session
Transaction
Card
Withdrawal
Deposit
Transfer
Inquiry
Account Information

State-Machine Diagrams for Example ATM System

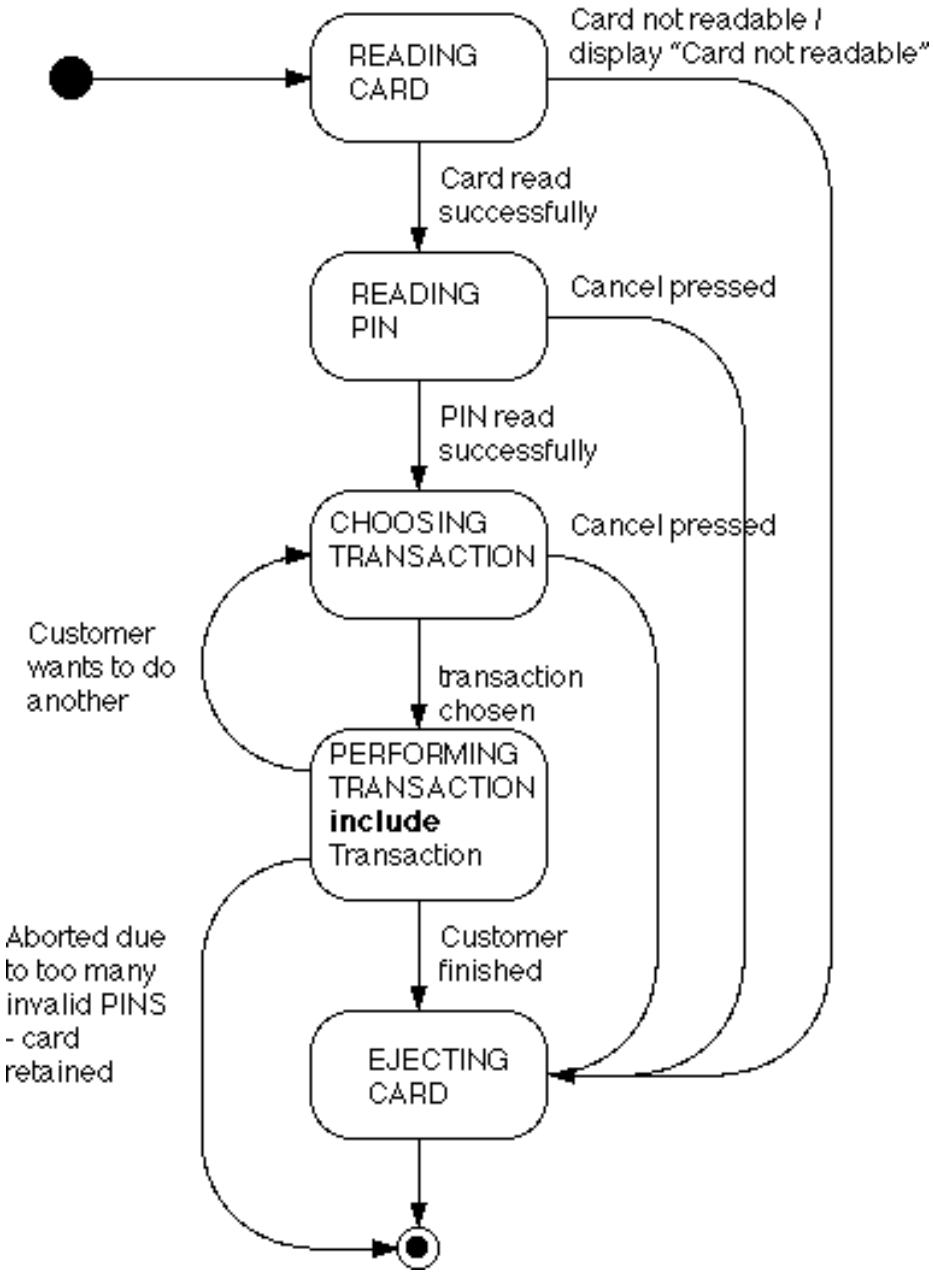
- Three of the objects we have identified have behavior that is sufficiently complex to warrant developing a State Chart for them. (These are the objects that were identified as the major controller objects.)
  - The object representing the machine itself (responsible for the System Startup and Shutdown use cases)
  - Objects representing a customer session (one per session) (responsible for the Session use case)
  - Objects representing an individual transaction (one per transaction) (responsible for the Transaction use case, use cases for the specific types of transaction, and Invalid PIN extension).

University of Bedfordshire

State-Chart for Overall ATM (includes System Startup and System Shutdown Use Cases)

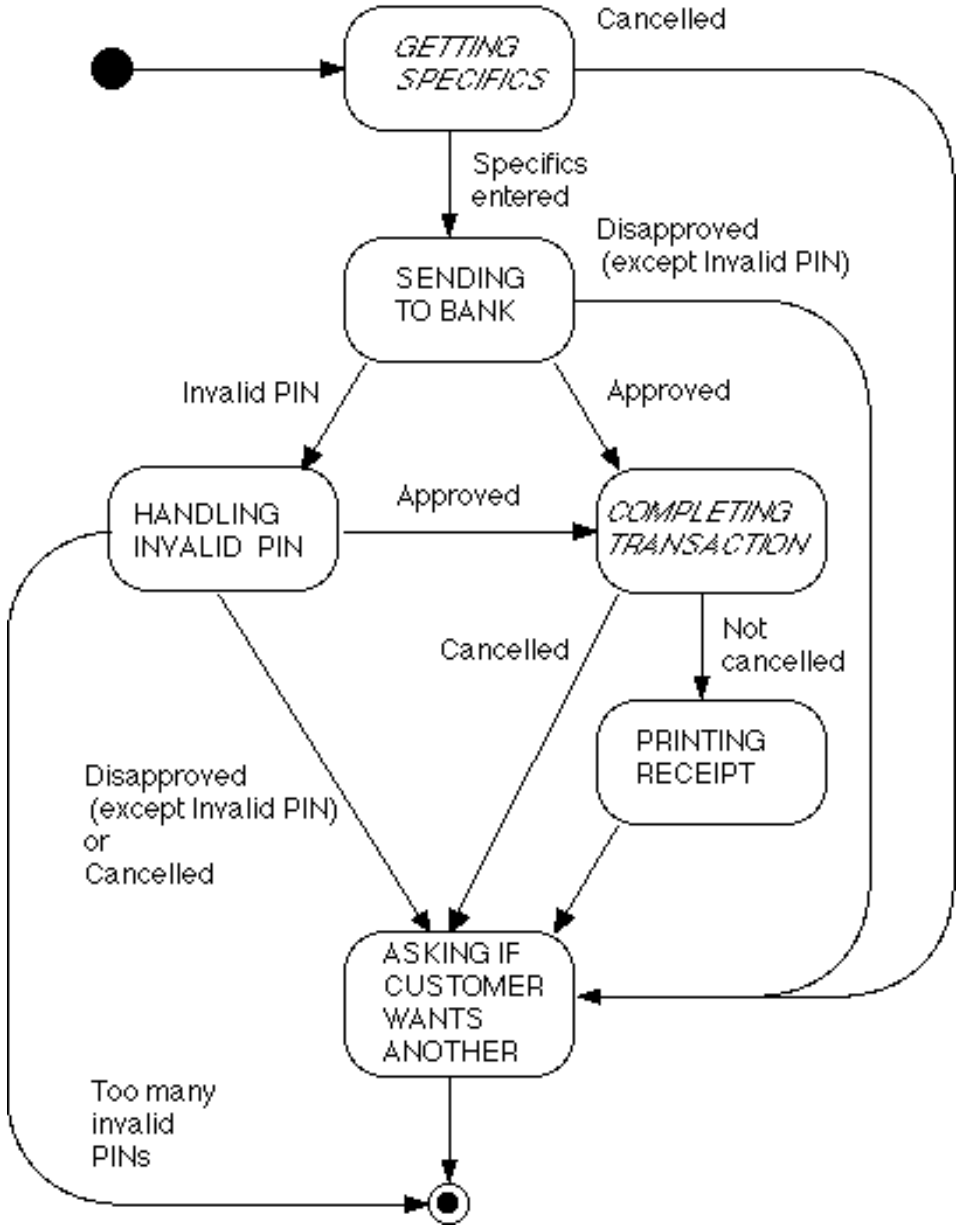


State-Chart for One Session



University of Bedfordshire

State-Chart for One Transaction (italicized operations are unique to each particular type of transaction)



Detailed Design

- The methods needed by each class are implicit in the responsibilities assigned to the class in the CRC cards, and become explicit in the Interaction Diagrams. A responsibility listed for a class on its CRC card generally maps into a method or methods in the detailed design. Likewise, any time an object belonging to a given class is shown as the recipient of a message in either a Sequence or Collaboration Diagram, the class needs a corresponding method. Many of the needed attributes are also either explicitly or implicitly present in the diagrams; the need for others becomes evident as the code for the class is being written. (Thus detailed design and coding are a "round trip" process - detailed design dictates coding, and coding leads to elaboration of the detailed design.)

Detailed Design

- In designing this system, a few key design decisions were made:
  - The class ATM is made an active class - that is, the ATM object has its own thread. Using the Java thread facllity leads to defining a run() method in this class whose body is executed by the ATM's thread. The fact that class ATM is active is indicated in class diagrams by enclosing it in a heavier outline.
  - Certain signals initiate computation - e.g. the signals from the operator console when the state of the switch is changed, or from the card reader when a card is inserted. In the GUI simulation of the ATM, these signals are sent by the "actionPerformed()" method of the appropriate GUI button; in a real ATM they would be sent by the physical components themselves, which might then also need to be active classes. (Note: this forms an exception to the rule that a responsibility on a CRC card translates into a method in the design - in this case the class sends a signal, rather than receiving it, so it does not need a method directly corresponding to the responsibility.)
  - The Transaction hierarchy consists of the abstract class Transaction and four concrete subclasses (Withdrawal, Deposit, Transfer and Inquiry). The class Transaction has a "virtual constructor" called makeTransaction() which asks the customer to choose a transaction type and then constructs and returns an object of the appropriate subclass. The Transaction class is made responsible for carrying out the Transaction use case and the Invalid PIN extension; for the former, it makes use of abstract methods getSpecificsFromCustomer() and completeTransaction() which are implemented concretely by each subclass.
  - The class Receipt is abstract. The completeTransaction() method of each kind of transaction creates a concrete instance that contains the information relevant to that kind of transaction.
  - The class Status is abstract. The send() method of NetworkToBank constructs a concrete instance that contains the information appropriate to the response received from the bank to a particular message.