COMP23420 Lecture 7

Behavioural Modelling

Kung-Kiu Lau
kung-kiu@cs.man.ac.uk
Office: Kilburn 2.68

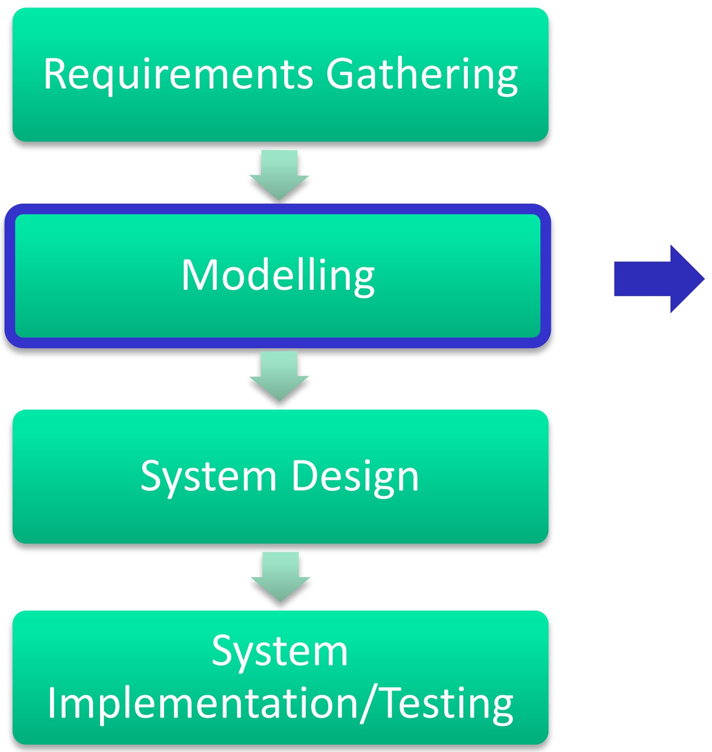
MANCHESTER
1824

The University of Manchester

Overview

- Where we are in the development process
- Adding behaviour to the structural model
- Sequence diagram
- Communication diagram
- State machine diagram
- Workshop 4

Where we are in the Development Process



Functional Model
- Activity Diagram
- Use Case Diagram

Structural Model
- Domain Model (Classes)
- System Classes

Behavioural Model
- Sequence Diagrams
- Communication Diagrams
- State Machine Diagrams

We have the structural model; and now we add behaviour to the structural model, by defining the interactions between the system classes.

Structure vs Behaviour

So far we have only defined structure
(structural model: domain classes, system classes)

Classes
- have only structure

Objects
(instances of classes)
- have behaviour

Class diagrams define:

- structure
- not behaviour

To add behaviour:

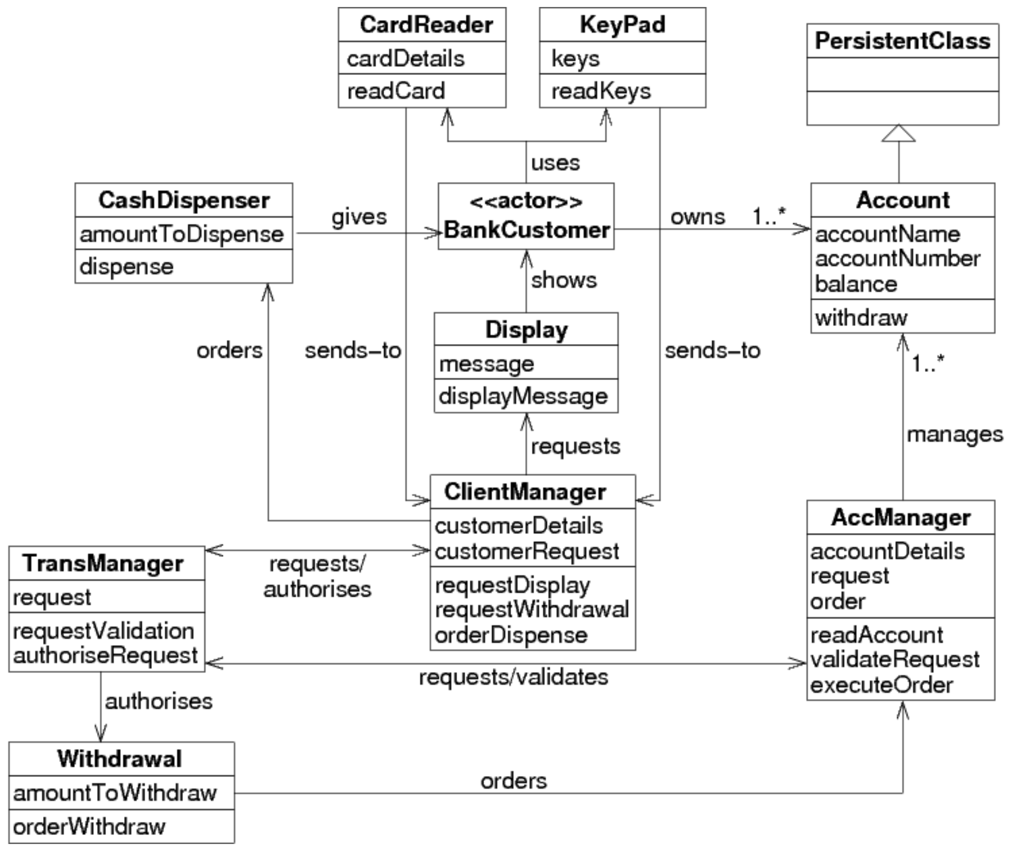
- define behaviour of corresponding objects and all their interactions

System Classes: The ATM Example

No behaviour in system classes

To add behaviour:

- define behaviour of all objects of system classes and all their interactions



System classes for the Withdraw Money use case realisation

Behavioural Modelling

To add behaviour to system classes, we draw:

- Interaction diagrams
  - Sequence diagrams
  - Communication diagrams
- State Machine diagrams

Interaction diagrams are object diagrams that specify interactions between objects

State machine diagrams specify the internal behaviour of single objects

With behavioural modelling, we are getting very close to code.

Specifying Behaviour for each Use Case Realisation

Need to make sure all use case realisations are covered

So specify behaviour for each use case realisation

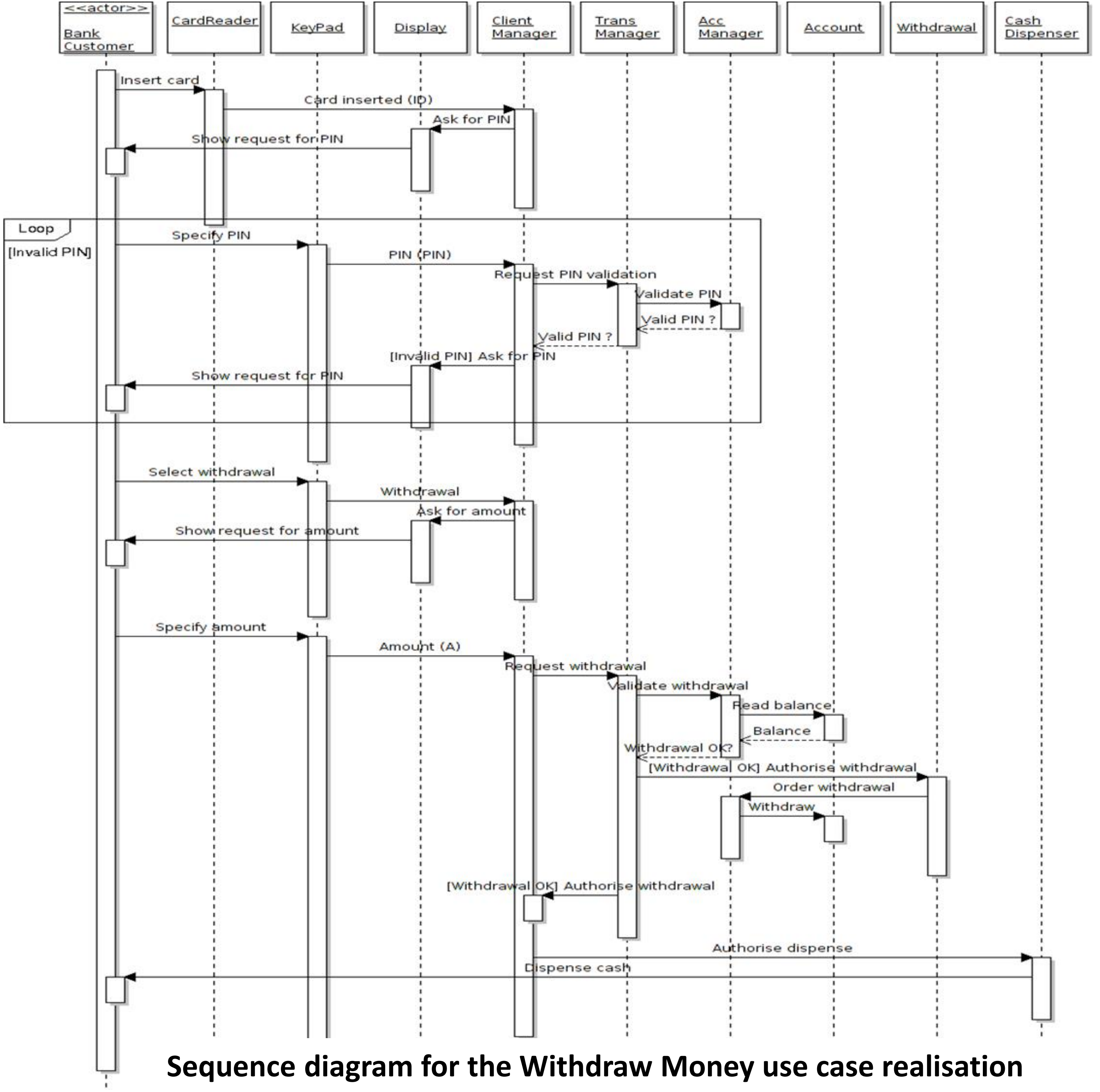
Behaviour of objects that collaborate in a use case realisation can be specified as a sequence of interactions between them

These interactions are messages and subsequent method executions

They can be defined by a sequence diagram or a communication diagram

The University of Manchester

Sequence Diagram: The ATM Example

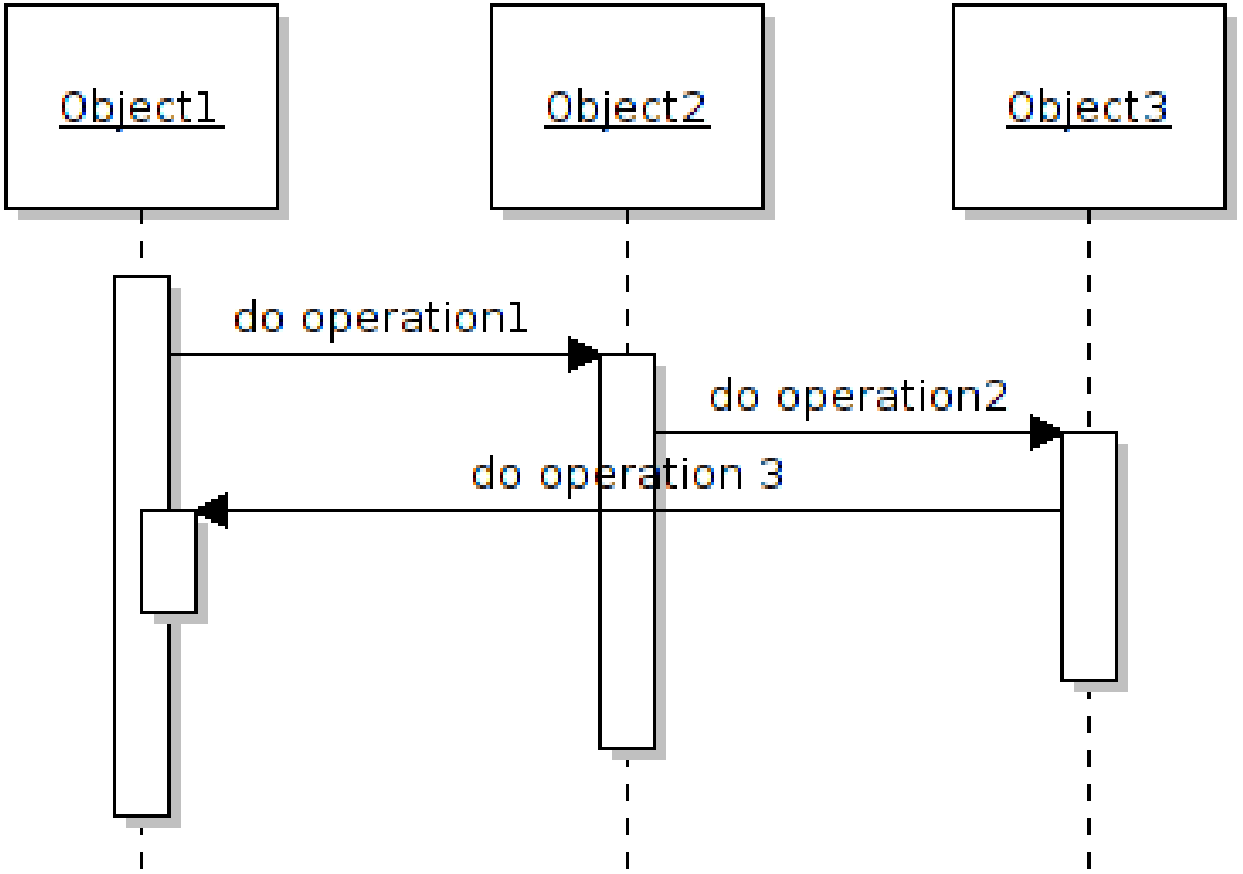


Sequence diagram for the Withdraw Money use case realisation

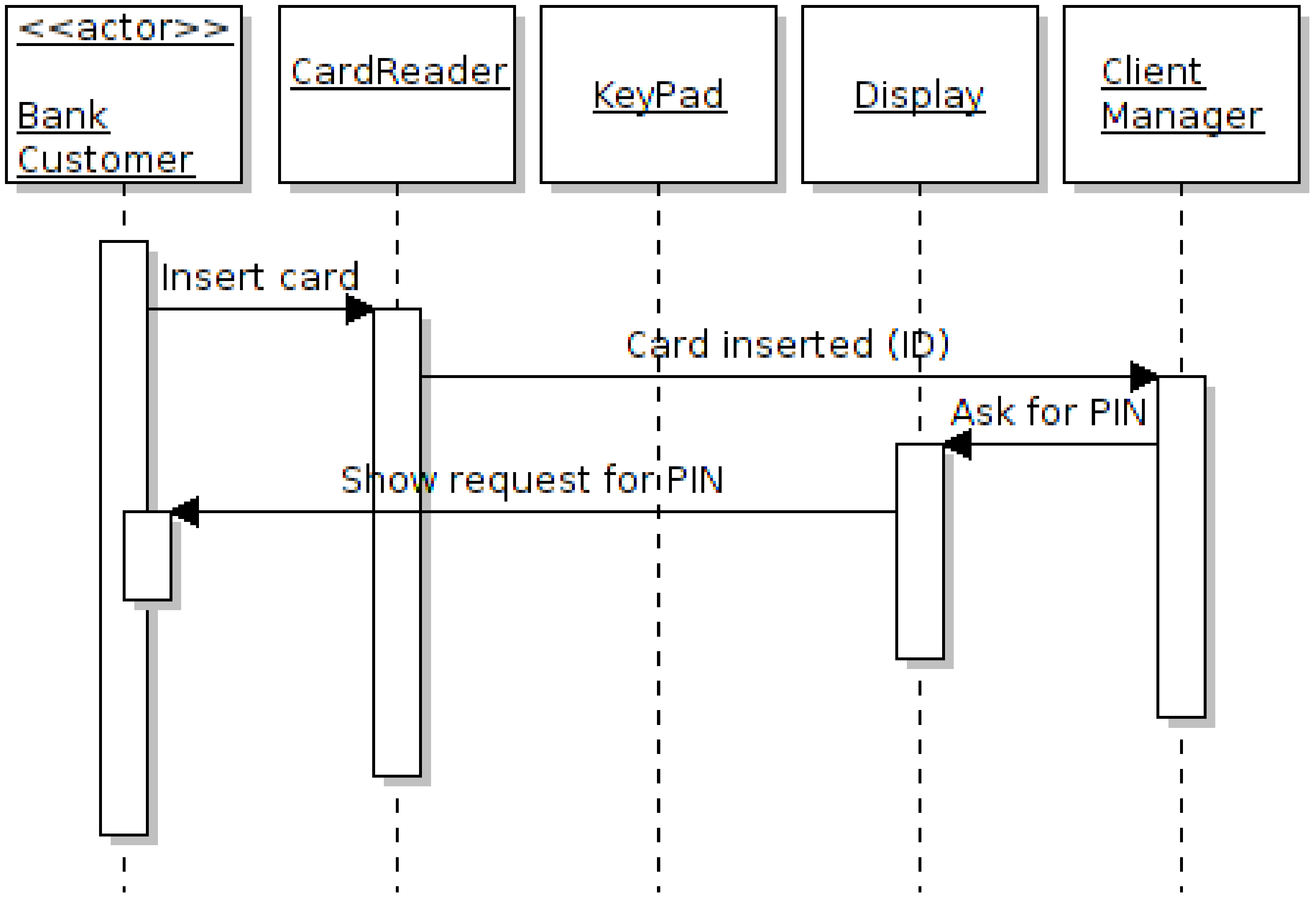
The University of Manchester

MANCHESTER 1824

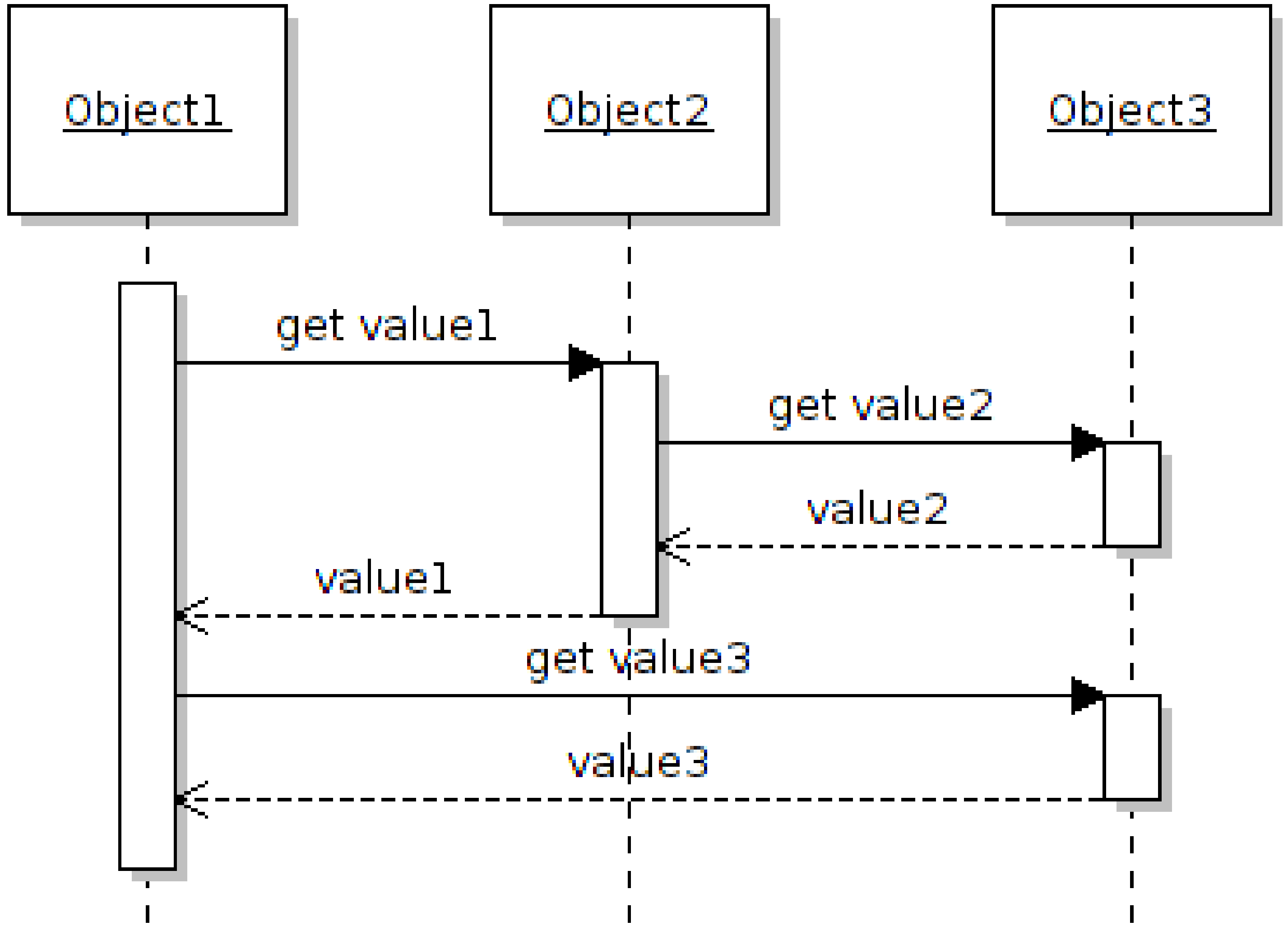
Sequence Diagrams: Sequencing Calls



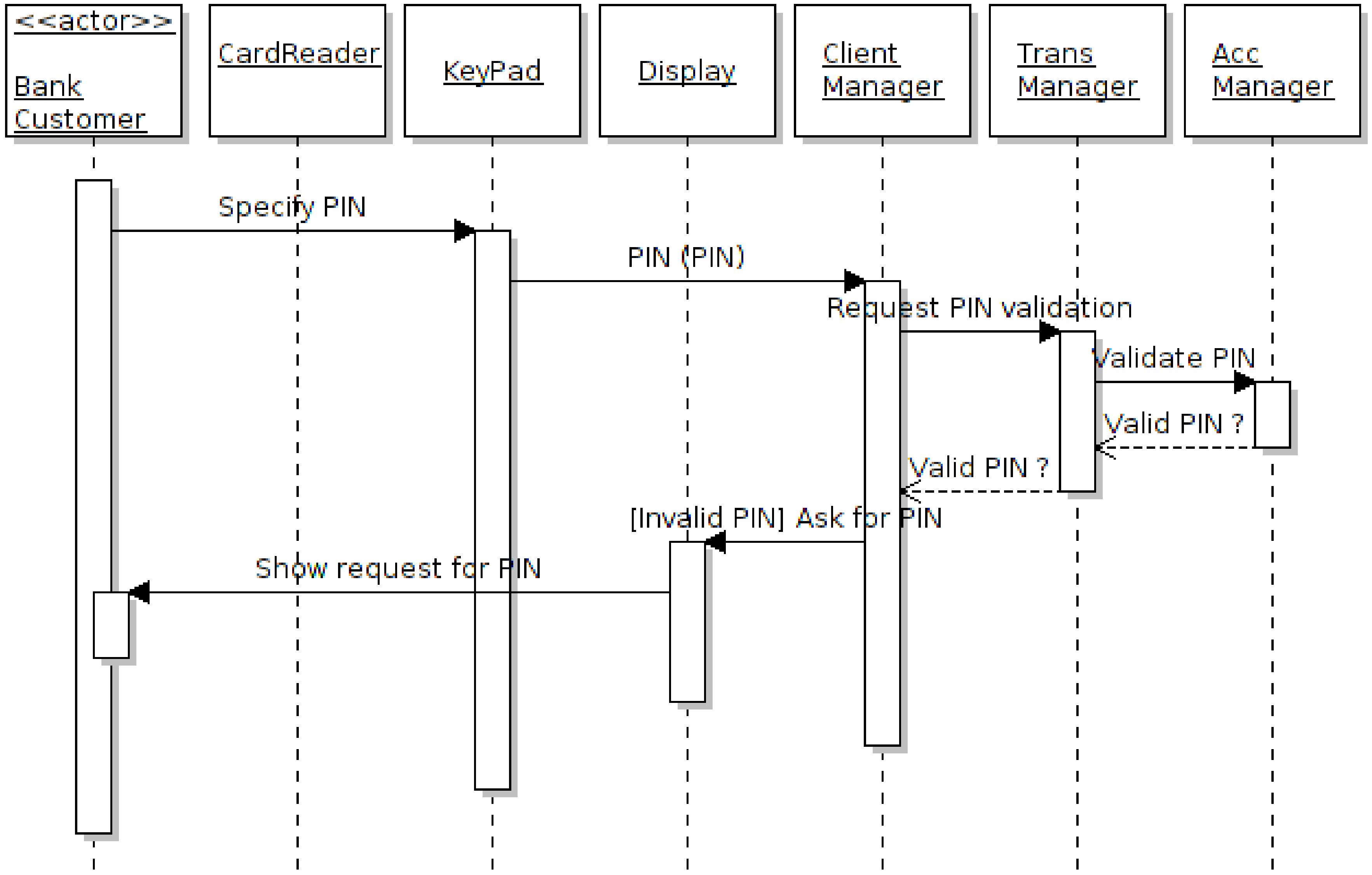
Sequence Diagrams: Sequencing Calls



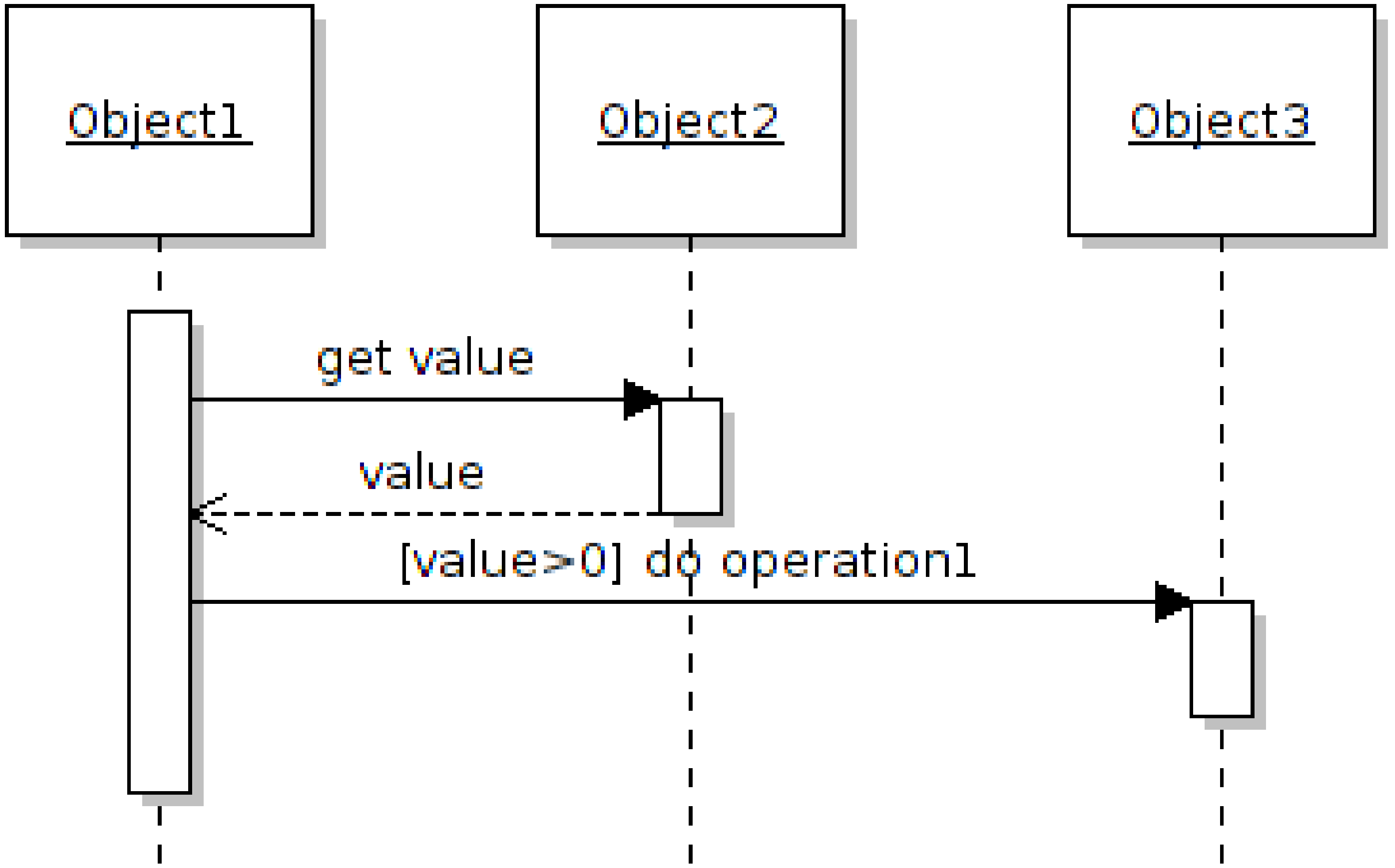
Sequence Diagrams: Returning Call Results



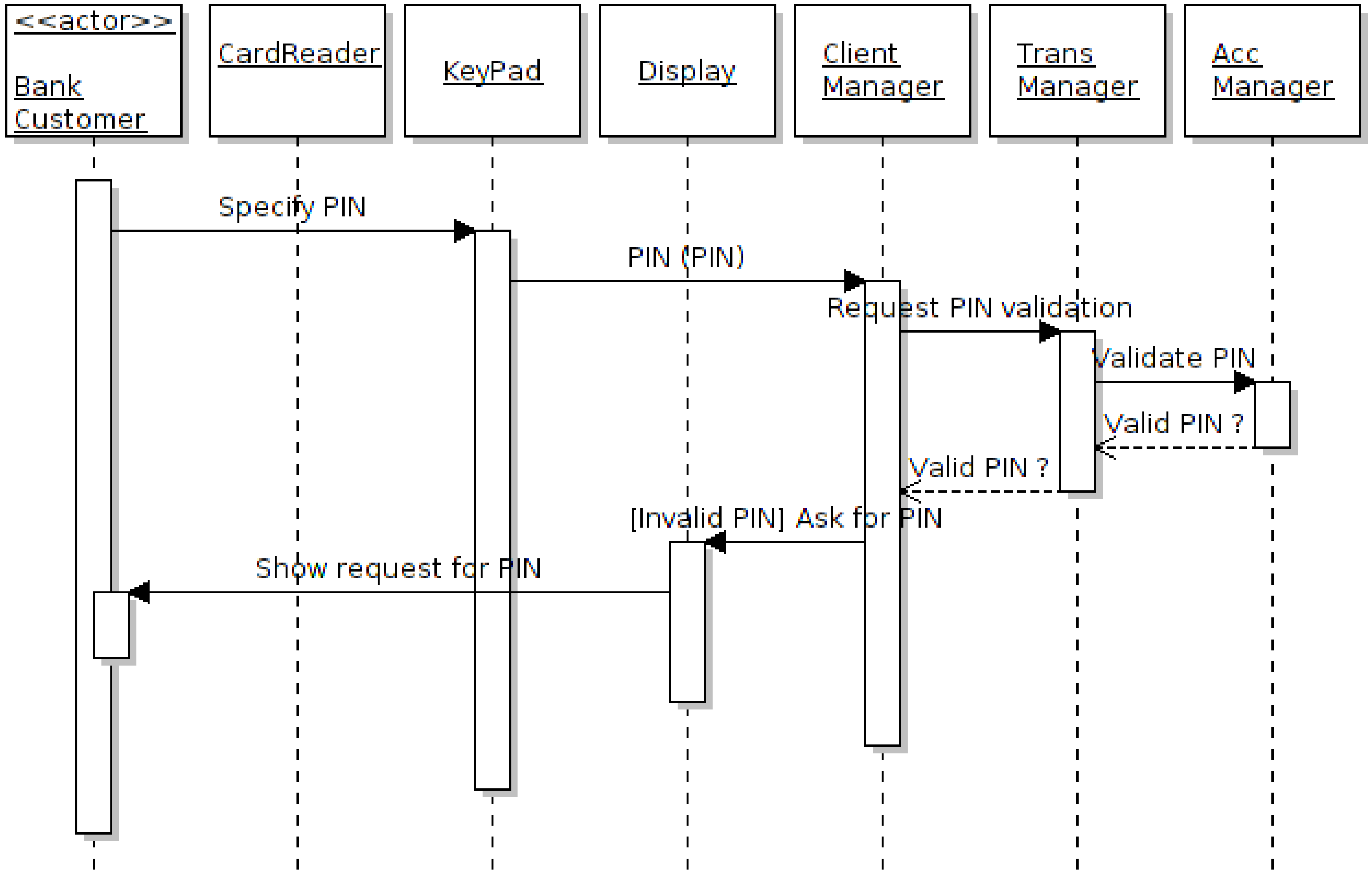
Sequence Diagrams: Returning Call Results



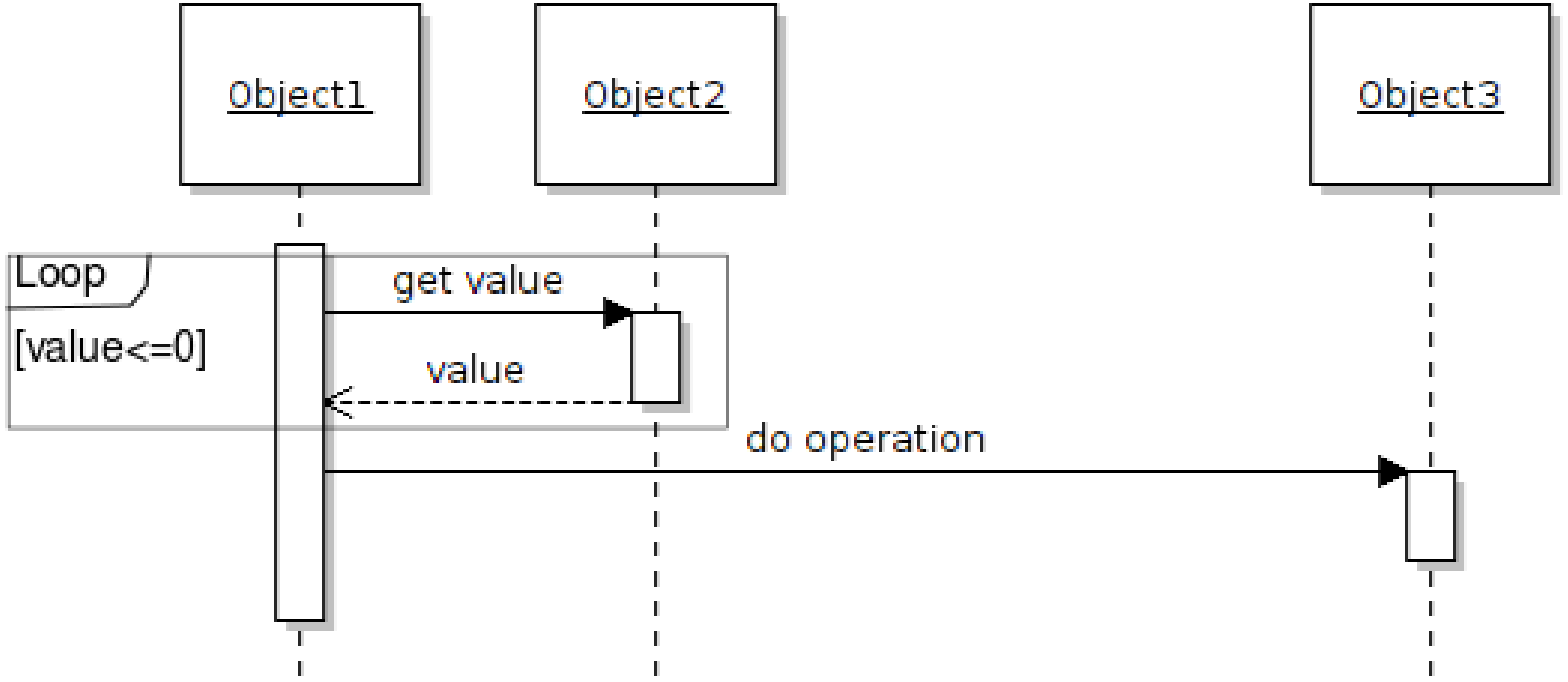
Sequence Diagrams: Conditional Calls



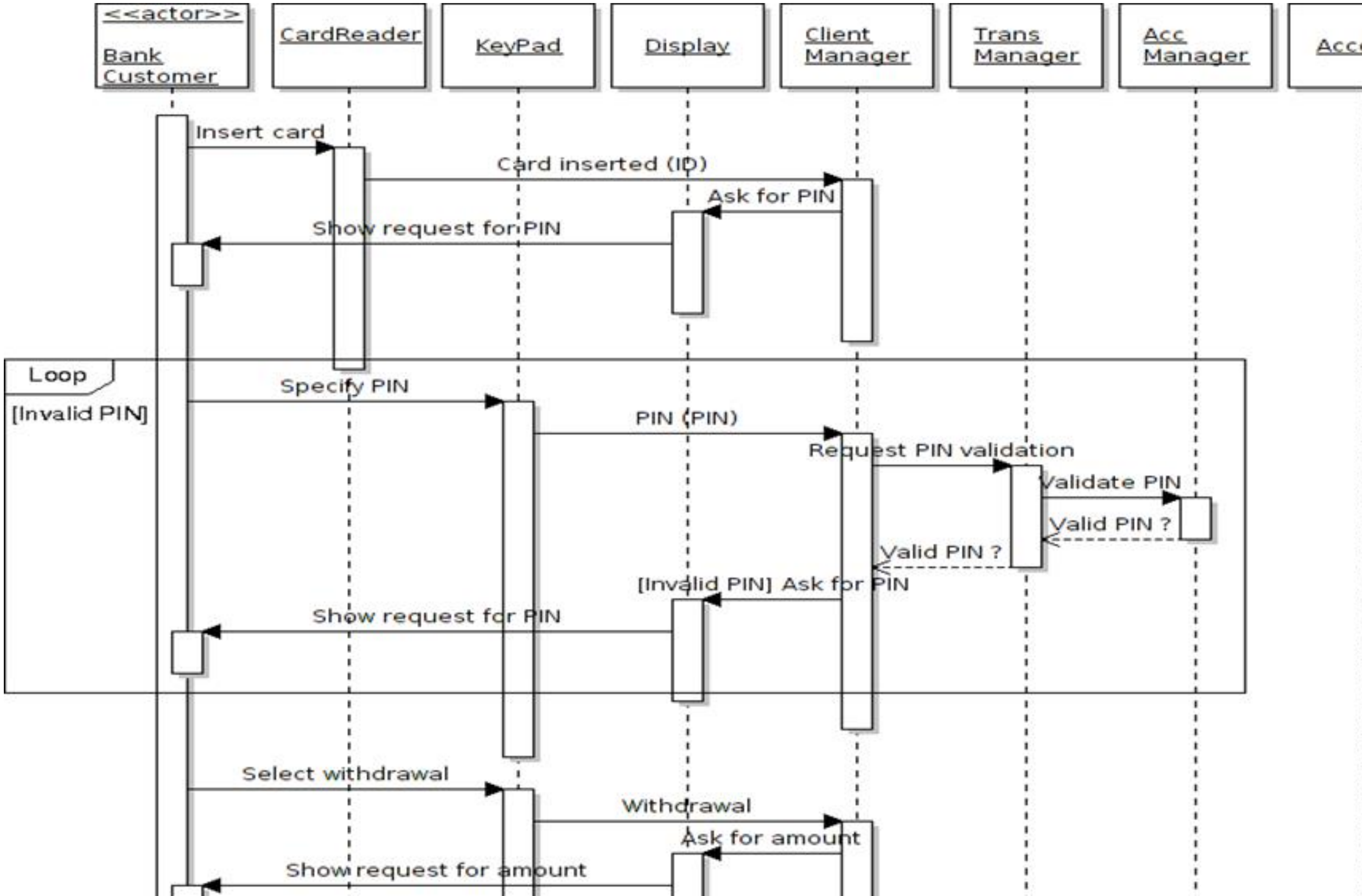
Sequence Diagrams: Conditional Calls



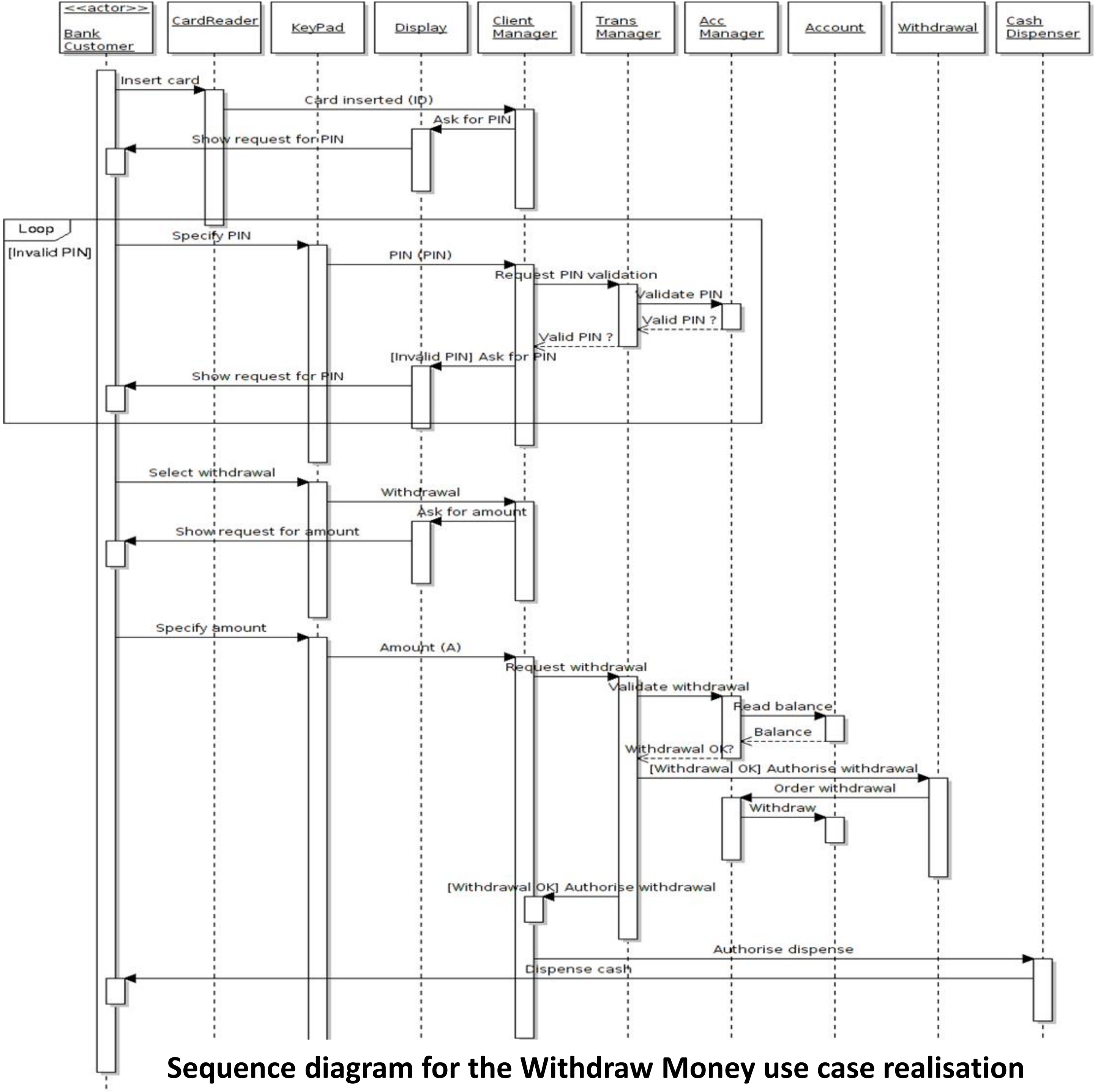
Sequence Diagrams: Loops



Sequence Diagrams: Loops



Sequence Diagram: The ATM Example



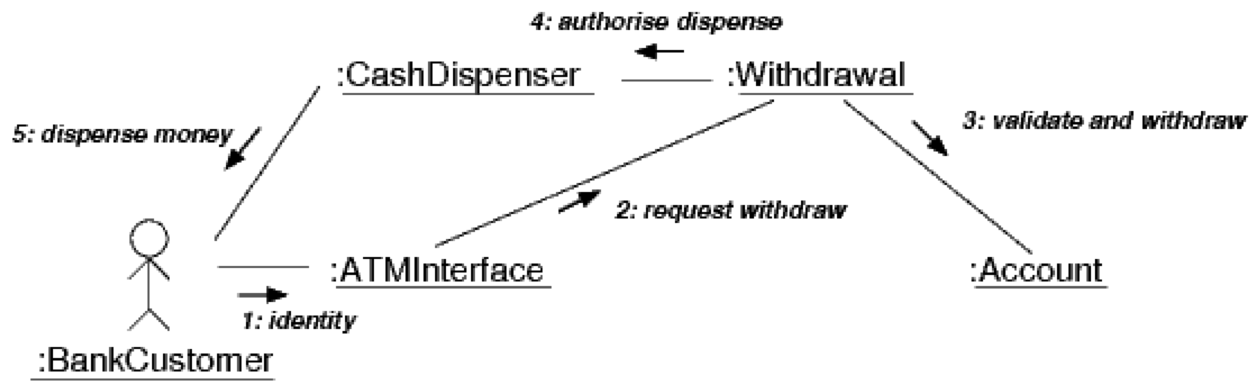
Sequence diagram for the Withdraw Money use case realisation

Communication Diagrams

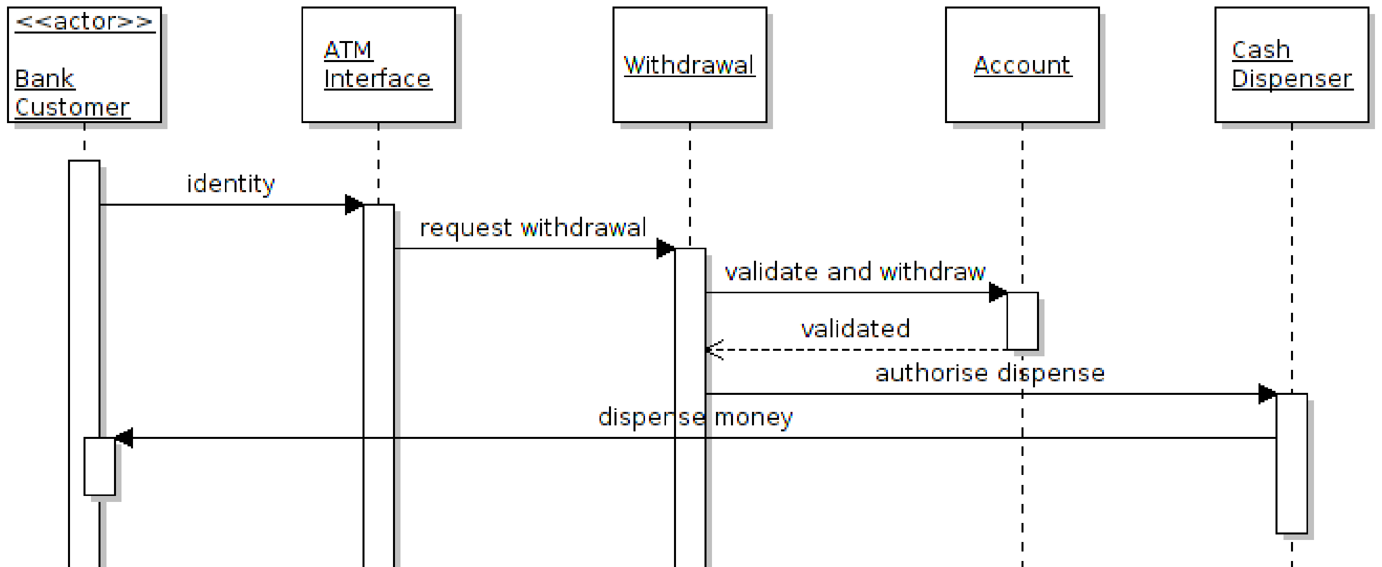
We could use communication diagrams instead.

They are equivalent to sequence diagrams.

For example, consider this communication diagram:

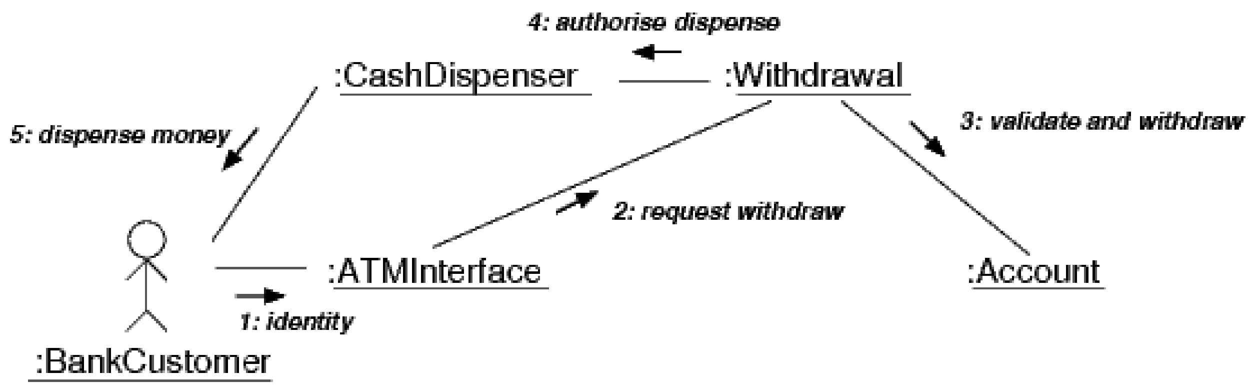


it is equivalent to this sequence diagram:

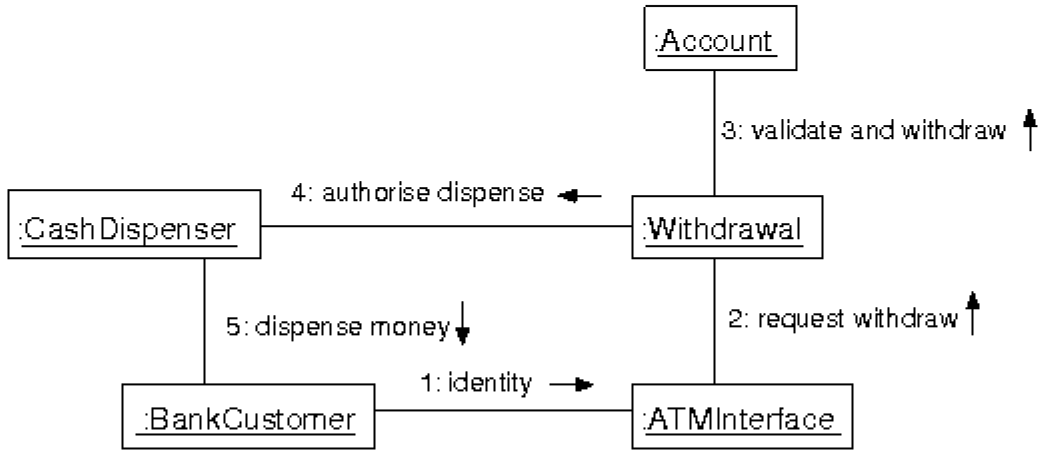


Communication Diagrams: Notation

No standard notation



A UML-like notation:



MANCHESTER 1824

The University of Manchester

Communication vs Sequence Diagrams

Sequence diagrams

- Show sequences of messages clearly
- Many notation options
- More complex
- Space consuming

Communication diagrams

- More difficult to see sequences of messages
- Fewer notation options
- Less complex
- Space efficient

MANCHESTER
1824

The University
of Manchester

Specifying the Internal Behaviour of a Single Object

We can add further behavioural details by specifying the internal behaviour of a single object

Regard an object as a state machine

A state machine has:

states

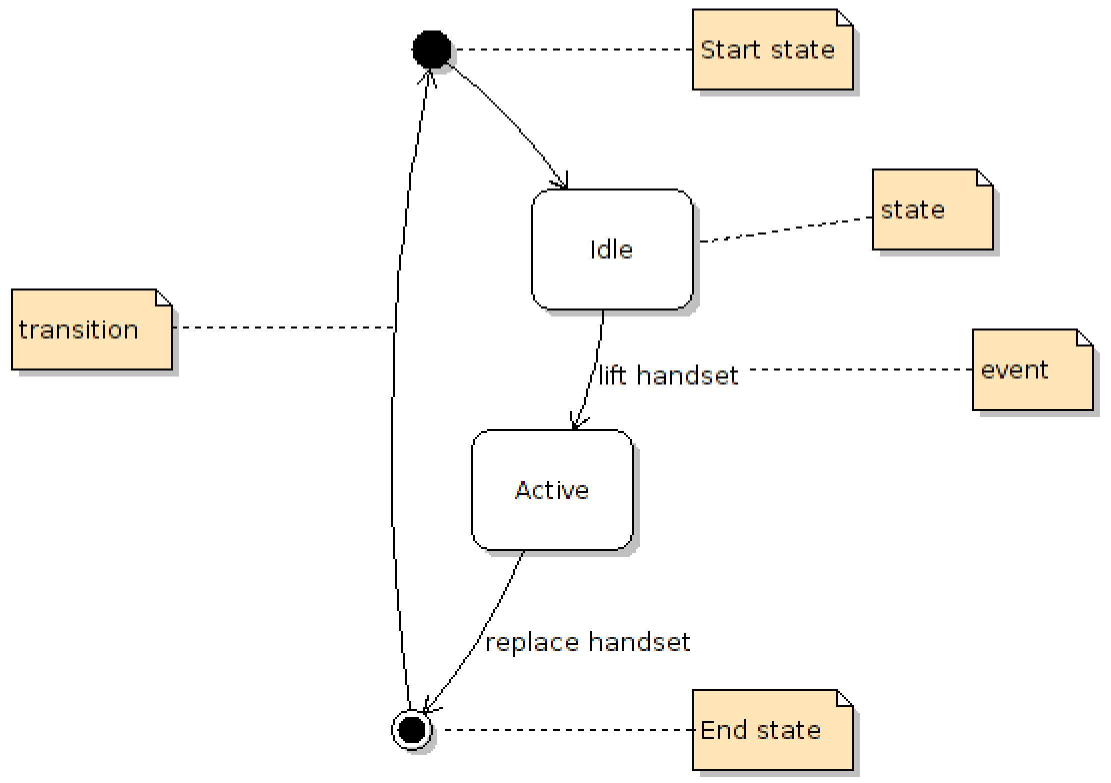
transitions between states

(triggered by events)

Defining an object's behaviour as a state machine gets very close to writing the code for that object

The University of Manchester

State Machine Diagrams: Example

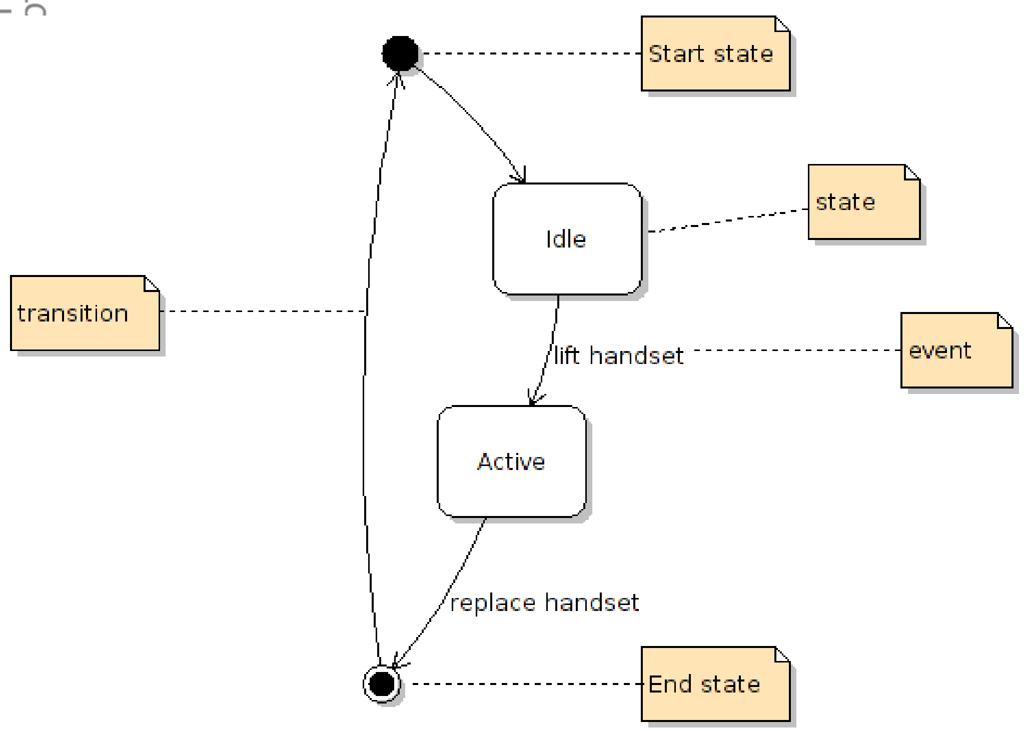


State machine diagram for a telephone

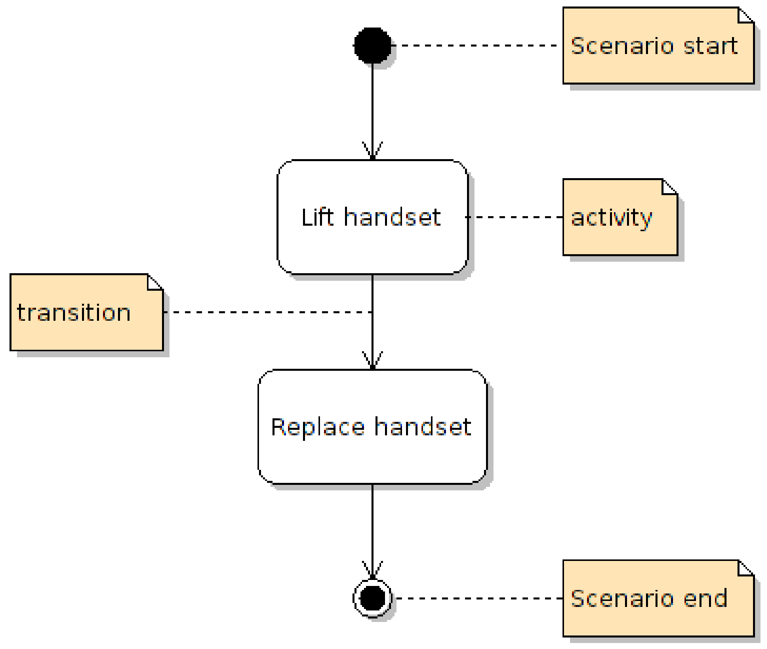
MANCHESTER 1824

The University of Manchester

State Machine Diagrams: Example



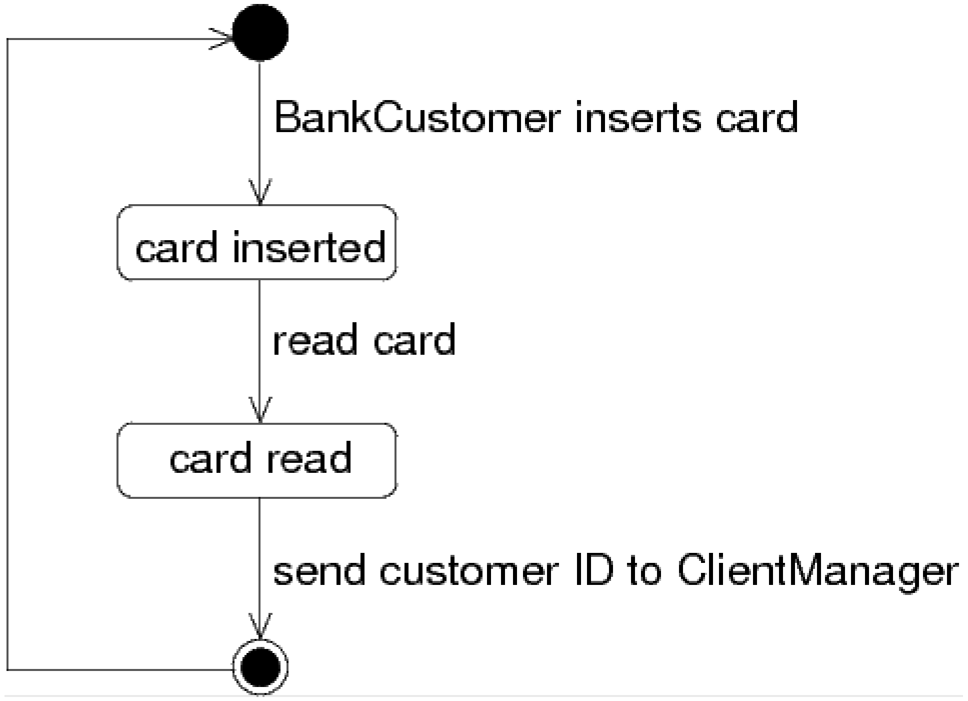
State machine diagram for an object



Activity diagram for a process

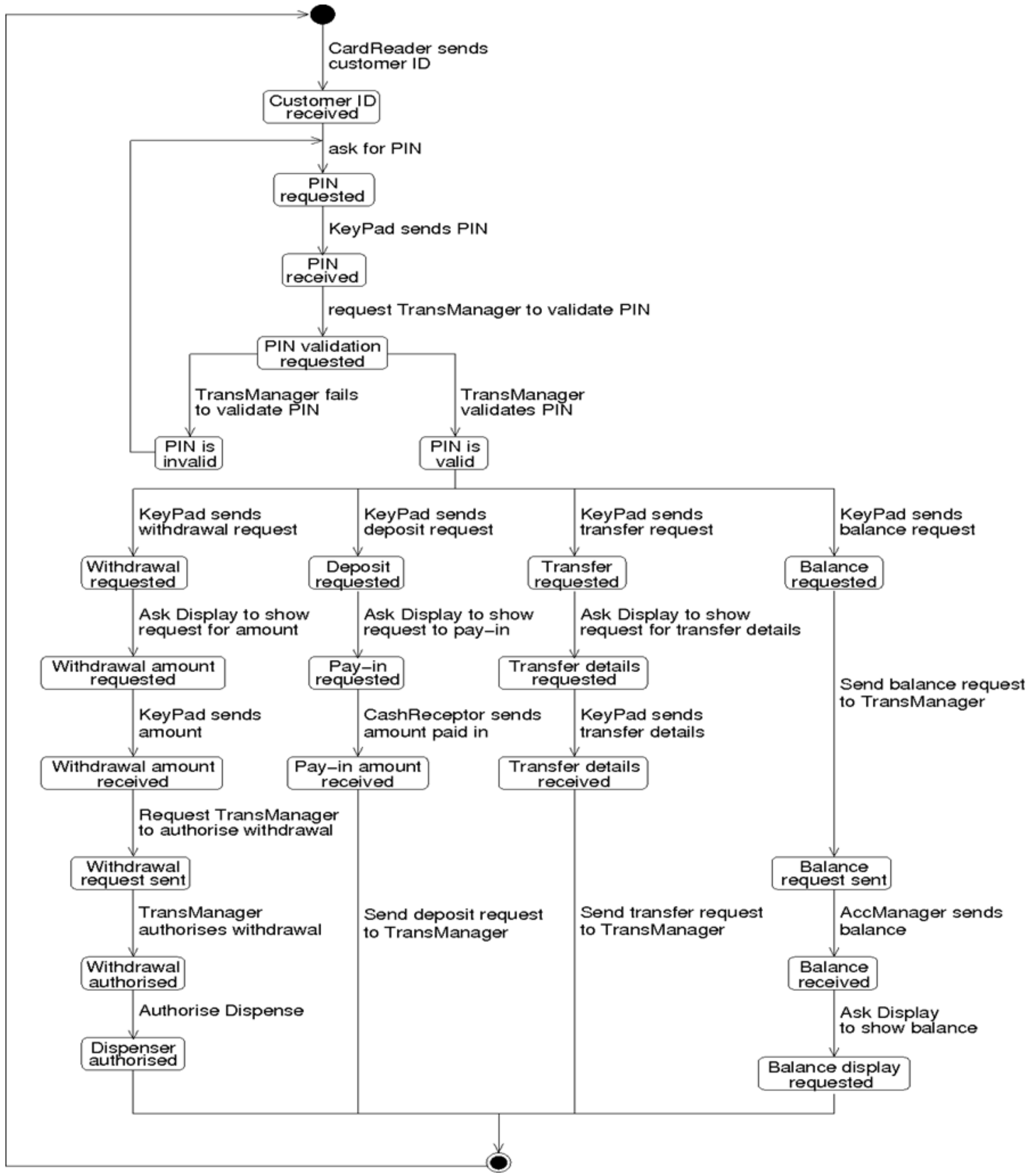
Do not confuse state machine diagrams with activity diagrams

State Machine Diagrams: ATM Example



State machine diagram for Card Reader

State machine diagram for ClientManager


CardReader sends customer ID
Customer ID received
ask for PIN
PIN requested
KeyPad sends PIN
PIN received
request TransManager to validate PIN
PIN validation requested
TransManager fails to validate PIN
TransManager validates PIN
PIN is invalid
PIN is valid
KeyPad sends withdrawal request
KeyPad sends deposit request
KeyPad sends transfer request
KeyPad sends balance request
Withdrawal requested
Deposit requested
Transfer requested
Balance requested
Ask Display to show request for amount
Ask Display to show request to pay–in
Ask Display to show request for transfer details
Send balance request to TransManager
Withdrawal amount requested
Pay–in requested
Transfer details requested
KeyPad sends amount
CashReceptor sends amount paid in
KeyPad sends transfer details
Withdrawal amount received
Pay–in amount received
Transfer details received
Request TransManager to authorise withdrawal
Withdrawal request sent
Balance request sent
TransManager authorises withdrawal
Send deposit request to TransManager
Send transfer request to TransManager
AccManager sends balance
Withdrawal authorised
Balance received
Authorise Dispense
Ask Display to show balance
Dispenser authorised
Balance display requested

Summary

Behavioural modelling adds behaviour to a structural model.

The behaviour of objects (that collaborate in a use case realisation) is specified as interactions between them.

These interactions can be specified by sequence diagrams or communication diagrams.

Sequence diagrams and communication diagrams are equivalent.

The internal behaviour of a single object is specified as a state machine.

MANCHESTER
1824

The University
of Manchester

Workshop 4: Behavioural Modelling for HTV

- Create sequence diagrams (for use cases)
- Create communication diagrams
- Create state machine diagrams

Bring:

- Laptops
  - For working
- USB sticks
  - For submission (feedback on Moodle later)
- System class diagram for HTV