OVERVIEW
BEHAVIOURAL MODELLING

What is it?

Why is it important?

Which behaviour is modelled?

Which is omitted?
WHAT IS BEHAVIOUR?

BEHAVIOUR IS INTERACTION BETWEEN OBJECTS.
UML DIAGRAMS
INTERACTION DIAGRAMS

Visualise the interactive behaviour of the system.

Describe the message flow in the system.

Describe structural organisation of the objects.

Describe interaction among objects.

Different types of models capture different aspects of the interaction.
KINDS OF DIAGRAMS

Use Case Diagrams (already considered)
Sequence Diagrams
Communication Diagrams
State Machine Diagrams
Activity Diagrams
Interaction Overview Diagrams
Timing Diagrams (not considered)
SEQUENCE DIAGRAMS
SEQUENCE DIAGRAM

Defines sequence of events, in particular, order in which messages occur

A natural refinement of Use Case diagrams
essentially an if statement
FRAMES: ALTERNATIVES

if-then-else
FRAMES: LOOPS

frames can be nested

also par (parallel) and region (critical sections)
SOME OTHER FEATURES

ClockStarter

startClock

Clock

create

run

selfCall

creation

active object

asynchronous

self call
GUIDELINES
Novice modellers do not pay enough attention to interaction diagrams

Do dynamic object modelling with interaction diagrams, not just static modelling with class diagrams.

Without modelling behaviour, you cannot really understand the system.

Interaction Diagrams can express use cases in concrete form that relates to classes
GUIDELINES

Walkthrough dynamic models against use cases.

Adjust static models whenever dynamic models expose weaknesses.

Ensure reuse among different dynamic models touching the same class – e.g., reusing a method for different scenarios.
ACTIVITY
MODEL THE FOLLOWING

A. player finding an opponent
B. initialising a game
C. pushing play after placing word
D. finishing a game (when final move is made)
E. selecting and continuing an ongoing game
OTHER BEHAVIOURAL MODELS
COMMUNICATION DIAGRAMS
COMMUNICATION DIAGRAM

illustrate object interactions in a graph or network format
objects are placed anywhere in the diagram
captures essence of wall sketch
EXAMPLE

direction of flow

makePayment(cashTendered)  
:Register

sequence number

1:makePayment(cashTendered)  
:Sale

nesting – subcall of 1

1.1:create(cashTendered)  
:Payment

conditionals also possible
LARGER EXAMPLE
COMPARISON
SEQUENCE VS COMMUNICATION DIAGRAMS

Sequence diagrams:

- easier to see flow, just follow arrows (vs numbers)
- large number of notation options
- rigid structure – consumes horizontal space

Communication diagrams:

- easier to sketch free form, more space efficient, and easier to modify on the fly
- difficult to see sequence of messages
- fewer notation options
STATE MACHINE DIAGRAM

illustrates the behaviour of an object in reaction to an event based on interesting events and states of an object

show the lifecycle of an object

use only for state-dependent objects

limited use in business information systems

common in process control, device control, protocol handlers, and telecommunications
EXAMPLE

initial pseudo-state

state

start

transition

Idle

on hook

off hook

Active

on hook
GUARDS AND ACTIONS

Idle

off hook / play dial tone
[valid subscriber]

Active

action
(fires when transition is taken)

don hook
(guard (inhibits transition))
NESTED STATES

Idle

off hook / play dial tone

[valid subscriber]

on hook

Dialing

digit

digit

digit

digit

complete

Connecting

Connecting

complete

Talking

Talking

connected

Active

Active

PlayingDialTone

PlayingDialTone
INTERACTION OVERVIEW

DIAGRAM

Provide a big-picture overview of how a set of interaction diagrams are related in terms of logic and process flow.
FRAMES

Encloses some kind of interaction diagram. Name describe relevant component.
An interaction defined elsewhere can be referred to inside an Interaction Overview diagram.
EXAMPLE 1
ACTIVITY DIAGRAMS
ACTIVITY DIAGRAM

rich notation to show a sequence of activities, including parallel activities.

popular for business workflows and processes, and use cases.

partitioned into different actors

expresses control flow, but not actual message contents.

can incorporate external systems (e.g., databases)
EXAMPLE

partitions. different parties involved

object node. object produced or used by actions

EXAMPLE

object node. object produced or used by actions

EXAMPLE

object node. object produced or used by actions

EXAMPLE

object node. object produced or used by actions
ANOTHER EXAMPLE

Customer

Shop and Fill Cart

Cart

Cashier

Enter Cart Items

Calculate Taxes and Discounts

[ cash payment ]

Create Receipt

NextGen POS

Authorisation Service

Submit Authorisation Request

Authorise Payment

[ else ]