Information Systems

Analysis – III

Behavioural Modelling

Module SITS code: COIY059H7
Dr Brian Gannon

Recap

- **Objects**: an instantiation of a class
  - `Patient` is a class
  - `PhilipMarlowe` is an object (instantiation of the patient class)

- **Attributes**: characteristics of a class (e.g. patient name, patient address, phone…)

- **Operations**: the behaviors of a class, or an action that an object can perform

- **Messages**: information sent to objects to tell them to execute one of their behaviors

Introduction

- Systems have static and dynamic characteristics
- **Structural** models describe the **static** aspects of the system
- **Behavioral models** provide a **dynamic** view of the system
- Behavioral models describe how the classes described in the structural models interact in support of the use cases

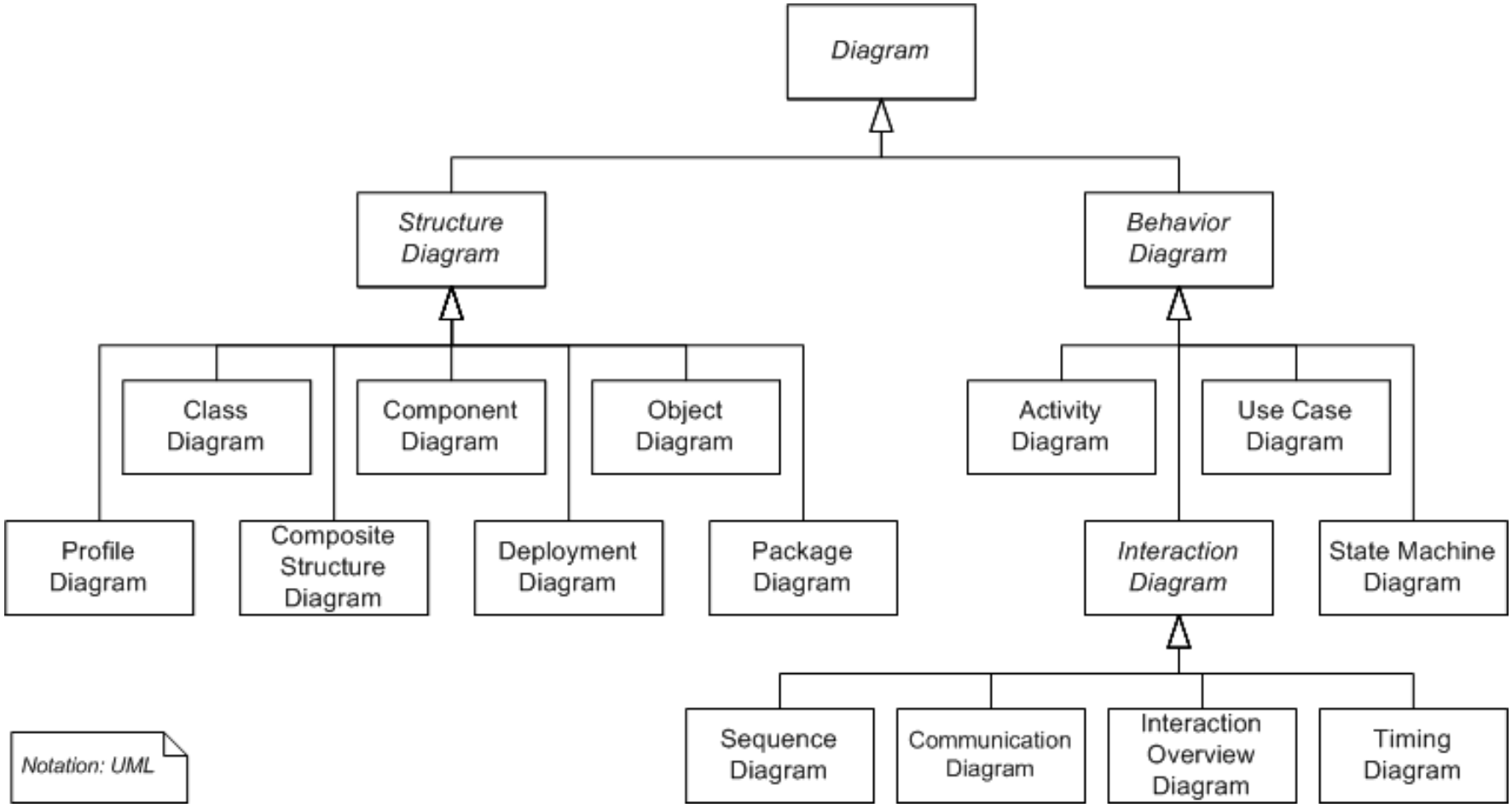
Behavioural Models

- Key UML behavioral models are **Behavioural State Machine** diagrams and Interaction diagrams (**Sequence** and **Communications** diagrams)

- Interaction diagrams show how objects interact to provide functionality defined in the use cases

- Behavioral state machines show how data change throughout the process - a dynamic model that shows the different states through which a single object passes during its life in response to events, along with its responses and actions

Modelling - UML Diagram Hierarchy



Paulo Merson: https://commons.wikimedia.org/wiki/File%3AUml_diagram2.png

Behavioural Models

- Key UML behavioral models are **Behavioural State Machine** diagrams and Interaction diagrams (Sequence and Communications diagrams)

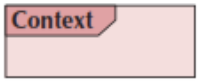
Behavioural State Machines

- Objects may change state in response to an event
- State machines capture the initial and changed states for objects
  - Show the different states through which a single object passes during its life
  - May include the object's responses and actions
- `Patient` states could include:
  - `New`: patient has not yet been seen
  - `Current:` patient is now receiving treatment
  - `Former:` patient is no longer being seen or treated
- Typically used only for complex objects

Components of Behavioural State Machines

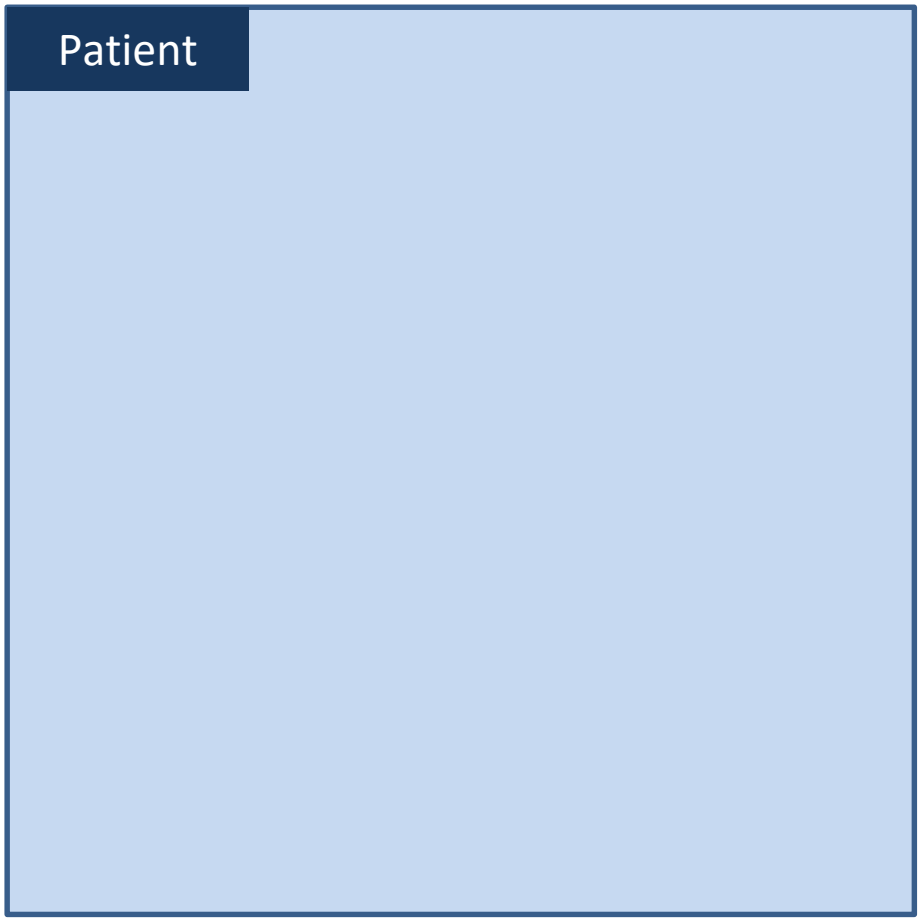
- **States**: values of an object's attributes at a point in time
- **Events**: the cause of the change in values of the object's attributes
- **Transitions**: movement of an object from one state to another
- **Conditions**: guard condition to flag that a condition is true and allow the transition
- **Actions**: atomic, non-decomposable processes
- **Activities**: non-atomic, decomposable processes

State Machine Syntax

Term and Definition	Symbol
A state: ■ Is shown as a rectangle with rounded corners. ■ Has a name that represents the state of an object.	
An initial state: ■ Is shown as a small, filled-in circle. ■ Represents the point at which an object begins to exist.	
A final state: ■ Is shown as a circle surrounding a small, filled-in circle (bull's-eye). ■ Represents the completion of activity.	
An event: ■ Is a noteworthy occurrence that triggers a change in state. ■ Can be a designated condition becoming true, the receipt of an explicit signal from one object to another, or the passage of a designated period of time. ■ Is used to label a transition.	anEvent
A transition: ■ Indicates that an object in the first state will enter the second state. ■ Is triggered by the occurrence of the event labeling the transition. ■ Is shown as a solid arrow from one state to another, labeled by the event name.	
A frame: ■ Indicates the context of the behavioral state machine.	

State Machine syntax

- Frame
  - Indicates the context of the state machine (an object)

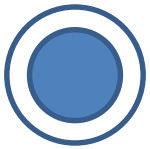


State Machine syntax

- Initial state
  - A small filled in circle
  - Represents the point at which an object begins to exist

State Machine syntax

- Final state
  - A small filled in circle surrounded by a circle
  - Represents the point at which an object no longer suffers activities in the current process

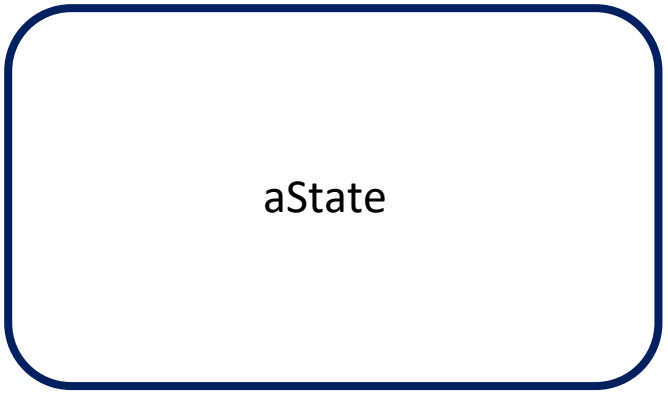


State Machine syntax

- Current state
  - A rectangle with rounded corners…

State Machine syntax

- Current state
  - A rectangle with rounded corners…
  - …with a name that represents a state of an object



State Machine syntax

- Transition
  - An arc which indicates that the state of the object at the tail will change to the state at the head

State Machine syntax

- Transition
  - An arc which indicates that the state of the object at the tail will change to the state at the head
  - …and is triggered by the **event** labelling the transition

State Machine syntax

- Event
  - A noteworthy occurrence triggering a change in state
  - May be receipt of a signal from another object

State Machine syntax

- Event
  - A noteworthy occurrence triggering a change in state
  - A designated condition becoming true

State Machine syntax

- Event
  - A noteworthy occurrence triggering a change in state
  - The passage of a designated period of time

State Machine syntax

- Event
  - An action or activity both shown after the slash (/)
  - **Action**: a single (atomic) process
  - **Activity**: a decomposable process (many actions)

[No stock]/re-order
Review re-order quantity

State Machine syntax

- So this is a dynamic diagram
  - A transition…

State Machine syntax

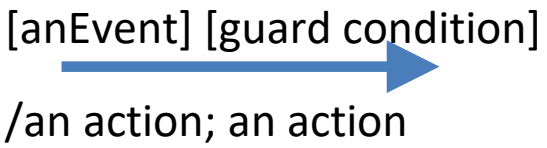
- So this is a dynamic diagram
  - A transition…
  - triggered by an event…

State Machine syntax

- So this is a dynamic diagram
  - A transition…
  - triggered by an event…
  - possibly guarded by a condition…

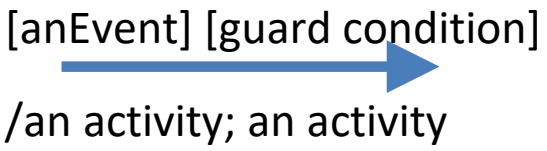
State Machine syntax

- So this is a dynamic diagram
  - A transition…
  - triggered by an event…
  - possibly guarded by a condition…
  - resulting in an action…

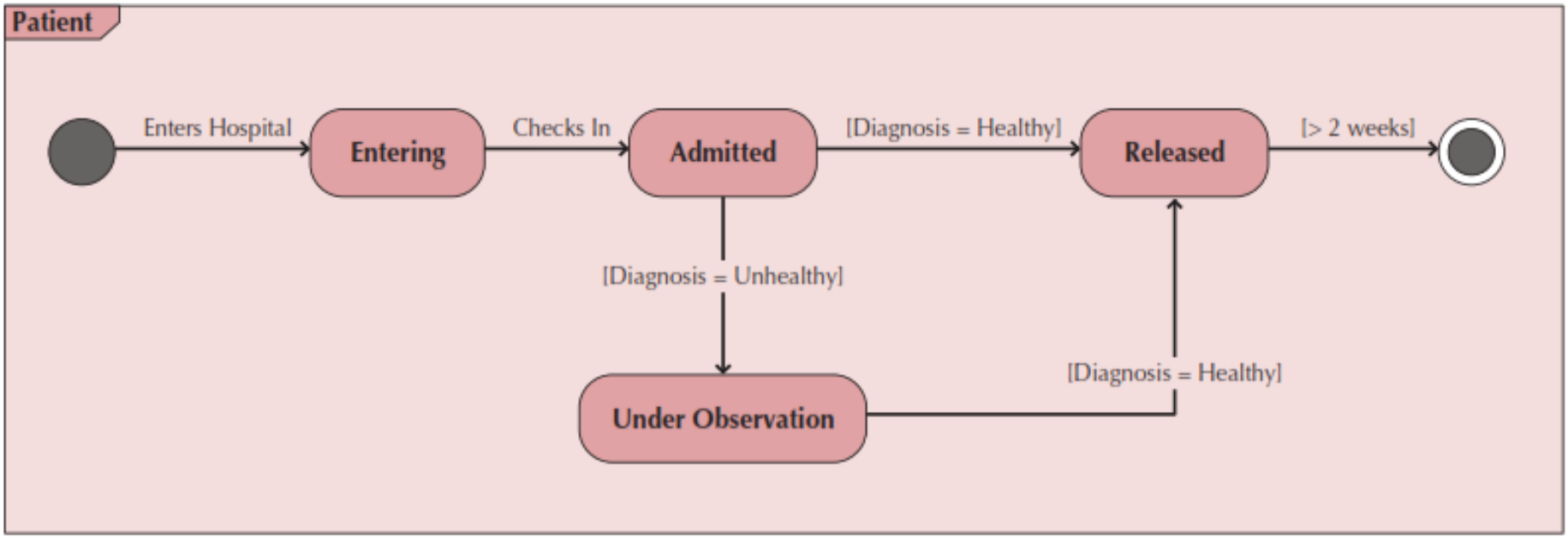


State Machine syntax

- So this is a dynamic diagram
  - A transition…
  - triggered by an event…
  - possibly guarded by a condition…
  - resulting in an action…
  - ..and/or activities



State Machine Example



Building State Machine Diagrams

- Set the context
- Identify the states of the object (Initial – Final – Stable)
- Lay out the diagram—use a ‘left to right’ or ‘top to bottom’ sequence
- Add the transitions
  - Identify the **events** (triggers that cause the transition)
  - Identify the **actions** which execute following a transition
  - Identify the **guard conditions** which determine outcome of transition
- Validate the model—ensure all states are reachable

Building State Machine Diagrams

- Use only for complex objects
- Draw the initial state in the upper left corner
- Draw the final state in the bottom right corner
- Use simple, but descriptive names for states
- Look out for “black holes” and “miracles”
- Ensure guard conditions are mutually exclusive
- Ensure transitions are associated with messages and operations

Behavioural Models

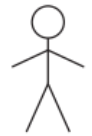
- Key UML behavioral models are Behavioural State Machine diagrams and Interaction diagrams (**Sequence** and Communications diagrams)

Sequence Diagrams

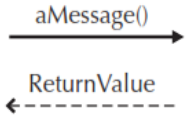
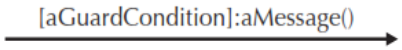
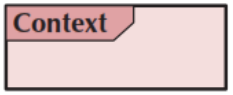
Sequence Diagram: emphasises message sequence

- Illustrate the objects that participate in a single use-case
- Dynamic model that shows sequence of messages that pass between objects
- Generic diagram shows all scenarios for a use-case
- Instance diagrams show a single scenario

Sequence Diagram Syntax

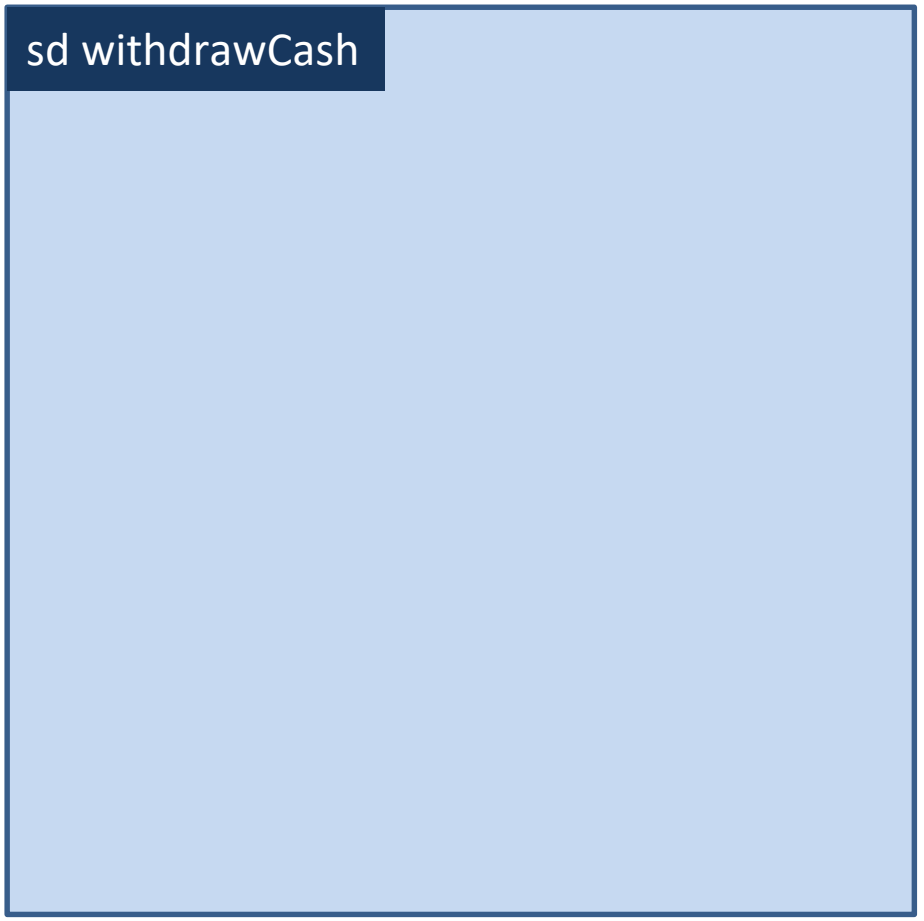
Term and Definition	Symbol
An actor: ■ Is a person or system that derives benefit from and is external to the system. ■ Participates in a sequence by sending and/or receiving messages. ■ Is placed across the top of the diagram. ■ Is depicted either as a stick figure (default) or, if a nonhuman actor is involved, as a rectangle with <<actor>> in it (alternative).	 anActor <<actor>> anActor
An object: ■ Participates in a sequence by sending and/or receiving messages. ■ Is placed across the top of the diagram.	anObject : aClass
A lifeline: ■ Denotes the life of an object during a sequence. ■ Contains an X at the point at which the class no longer interacts.	

Sequence Diagram Syntax

An execution occurrence: ■ Is a long narrow rectangle placed atop a lifeline. ■ Denotes when an object is sending or receiving messages.	
A message: ■ Conveys information from one object to another one. ■ A operation call is labeled with the message being sent and a solid arrow, whereas a return is labeled with the value being returned and shown as a dashed arrow.	
A guard condition: ■ Represents a test that must be met for the message to be sent.	
For object destruction: ■ An X is placed at the end of an object's lifeline to show that it is going out of existence.	X
A frame: ■ Indicates the context of the sequence diagram.	

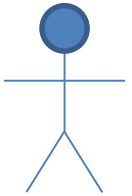
Sequence Diagram Syntax

- Frame
  - Indicates the context of the sequence diagram (a use case)



Sequence Diagram Syntax

- Actor
  - person or system external to the system and derives benefit from it
  - participates in a sequence by sending and/or receiving messages
  - placed across the top of the diagram
  - depicted either as a stick figure (default) or as a rectangle labelled <<actor>> (for non-human actors)



<<actor>>
Actor/Role

Sequence Diagram Syntax

- Object
  - participates in a sequence by sending and/or receiving messages
  - placed across the top of the diagram or placed at the point at which it is instantiated
  - depicted as a rectangle containing its name followed by the name of the class from which it is instantiated

Sequence Diagram Syntax

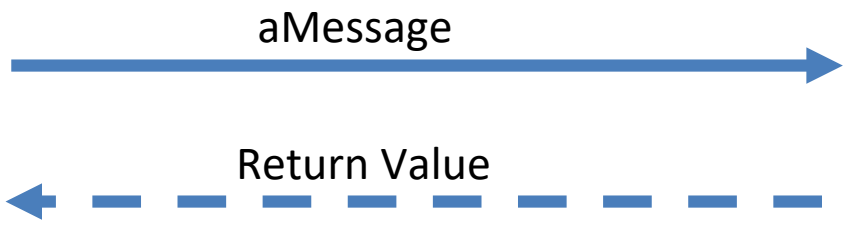
- Lifeline
  - denotes the life of an object during a sequence
  - depicted by a dashed line terminated by an X at the point at which the class no longer interacts

Sequence Diagram Syntax

- Execution occurrence
  - depicted by a long narrow rectangle placed over a lifeline
  - denotes when an object is sending or receiving messages

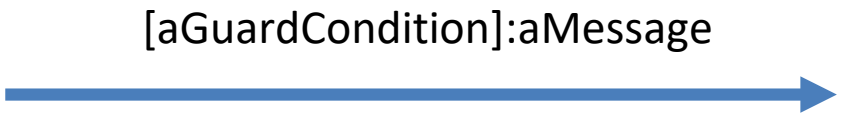
Sequence Diagram Syntax

- Message
  - conveys information from one object to another one
  - an operation call is shown as a solid arrow labelled with the message being sent
  - a return is shown as a dashed arrow labelled with the value being returned

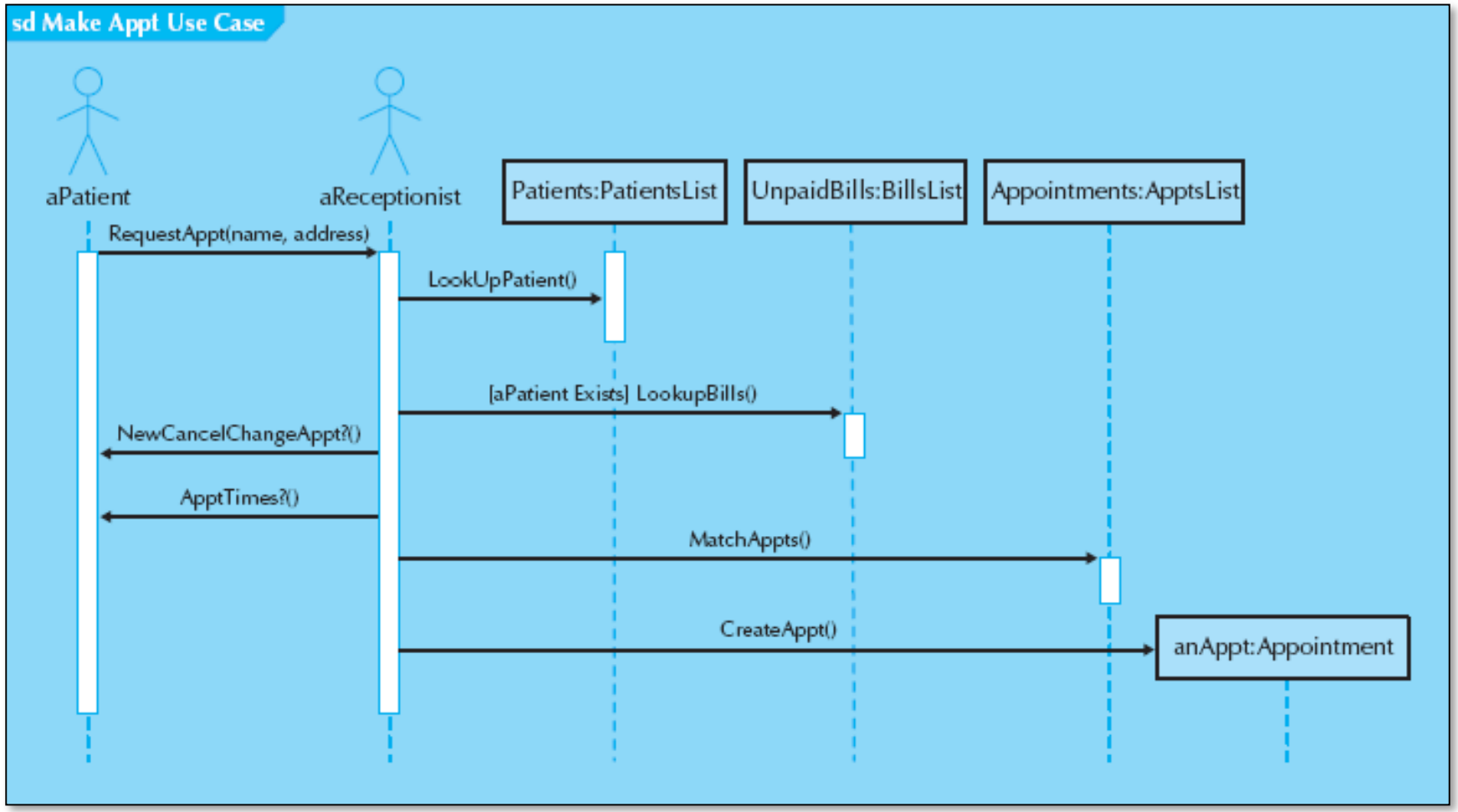


Sequence Diagram Syntax

- Guard condition
  - represents a test that must be verified as true for the message to be sent



Sequence Diagram Example



Building Sequence Diagrams

- Set the context
- Identify actors and objects that interact in the use-case scenario
- Set the lifeline for each object
- Add messages by drawing arrows
  - Show how they are passed from one object to another
  - Include any parameters in parentheses
  - Exclude obvious return values
- Add execution occurrence to each object's lifeline
- Validate the sequence diagram (ensure it depicts all steps in the process)

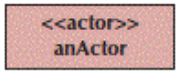
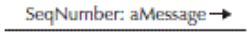
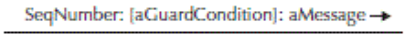
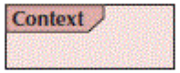
Behavioural Models

- Key UML behavioral models are Behavioural State Machine diagrams and Interaction diagrams (Sequence and **Communications** diagrams)

Communication Diagrams

- **Communication Diagram**: emphasises message flow
- An object diagram that shows message passing relationships
- Depicts the dependencies among the objects

Communication Diagram Syntax

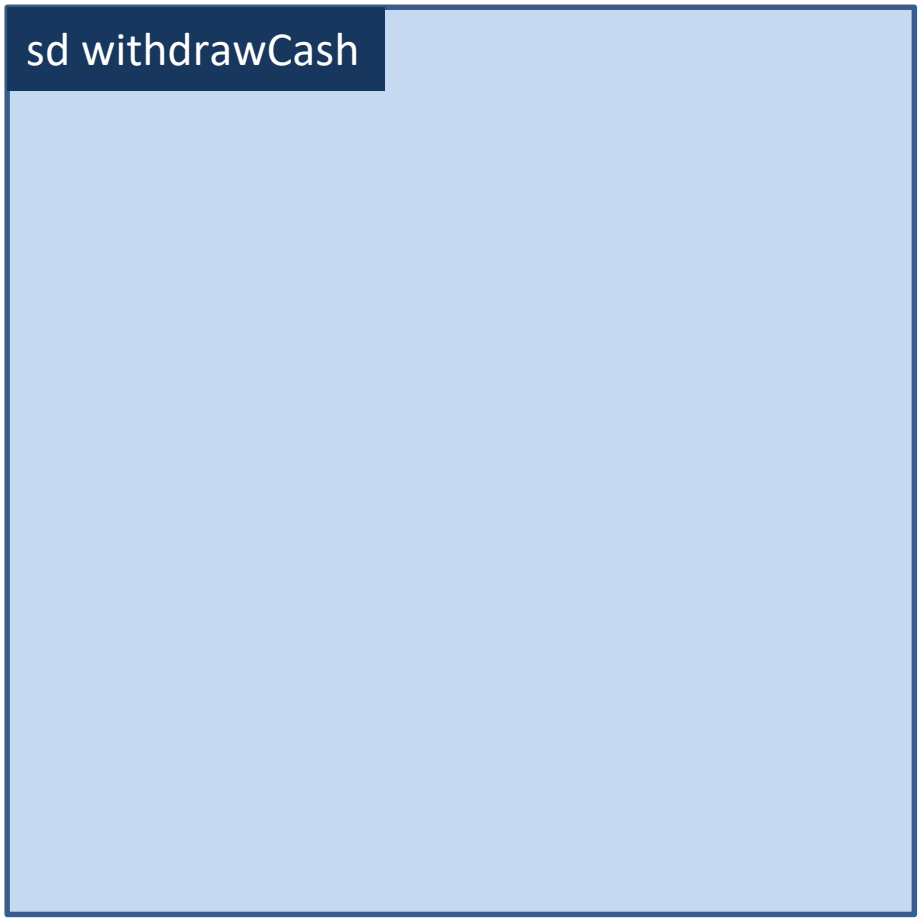
Term and Definition	Symbol
An actor: ■ Is a person or system that derives benefit from and is external to the system. ■ Participates in a collaboration by sending and/or receiving messages. ■ Is depicted either as a stick figure (default) or, if a nonhuman actor is involved, as a rectangle with <<actor>> in it (alternative).	 anActor 
An object: ■ Participates in a collaboration by sending and/or receiving messages.	
An association: ■ Shows an association between actors and/or objects. ■ Is used to send messages.	
A message: ■ Conveys information from one object to another one. ■ Has direction shown using an arrowhead. ■ Has sequence shown by a sequence number.	
A guard condition: ■ Represents a test that must be met for the message to be sent.	
A frame: ■ Indicates the context of the communication diagram.	

Communication Diagrams

- Illustrates the objects that participate in a use-case - but not with top-down sequence implication
- Shows the messages that pass between objects for a particular use-case, numbered in the sequence in which messages are passed
- Sequence diagram lays out the sequence of the process: Communications diagram is much more compact
- Both model the same thing and one may be transformed into the other.

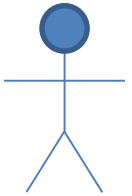
Communication Diagram Syntax

- Frame
  - Indicates the context of the communication diagram (a use case)



Communication Diagram Syntax

- Actor
  - person or system external to the system and derives benefit from it
  - participates in a sequence by sending and/or receiving messages
  - depicted either as a stick figure (default) or as a rectangle labelled <<actor>> (for non-human actors)



<<actor>>
Actor/Role

Communication Diagram Syntax

- Object
  - participates in a communication by sending and/or receiving messages
  - depicted as a rectangle containing its name followed by the name of the class from which it is instantiated

Communication Diagram Syntax

- Association
  - shows association between actors and/or objects
  - used to send messages

Communication Diagram Syntax

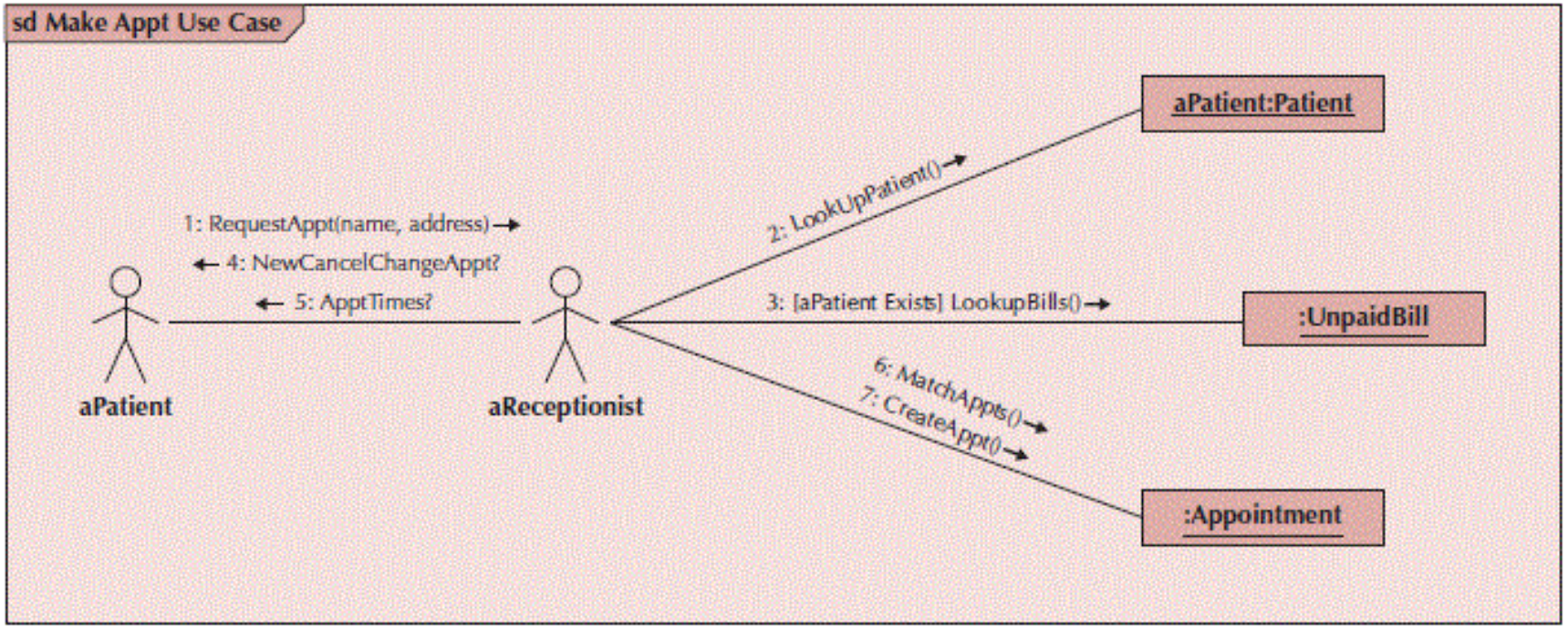
- Message
  - conveys information from one object to another one
  - sequence of message shown by a number
  - direction shown by an arrowhead

Communication Diagram Syntax

- Message
  - conveys information from one object to another one
  - sequence of message shown by a number
  - direction shown by an arrowhead
  - …may have a **guard condition** (test that must be met for the message to be sent)

SeqNumber:[guard Condition]aMessage

Communication Diagram Example



Building Communications Diagrams

- Set the context
- Identify objects, actors and associations between them
- Lay out the diagram
- Add the messages, numbered in sequence
- Validate the model against use cases

Validating Behavioural Models

- Actors must be consistent between models
- Messages on sequence diagrams must match associations on communication diagrams
- Every message on a sequence diagram must appear on an association in a communication diagram
- Guard conditions on a sequence diagram must appear on a communication diagram
- Sequence of messages must correspond to top down ordering of messages being sent
- State transitions must be associated with a message on a sequence diagram

References

Dr. BRIAN GANNON

E: b.gannon@bbk.ac.uk
E: brian.gannon@vesime.com
M: 07870 153855
LI: http://lnkd.in/QDXaH6
Tw: @bjgann