MSc Information Systems

Analysis – I

Business Process Modelling

Module SITS code: COIY059H7

Dr Brian Gannon

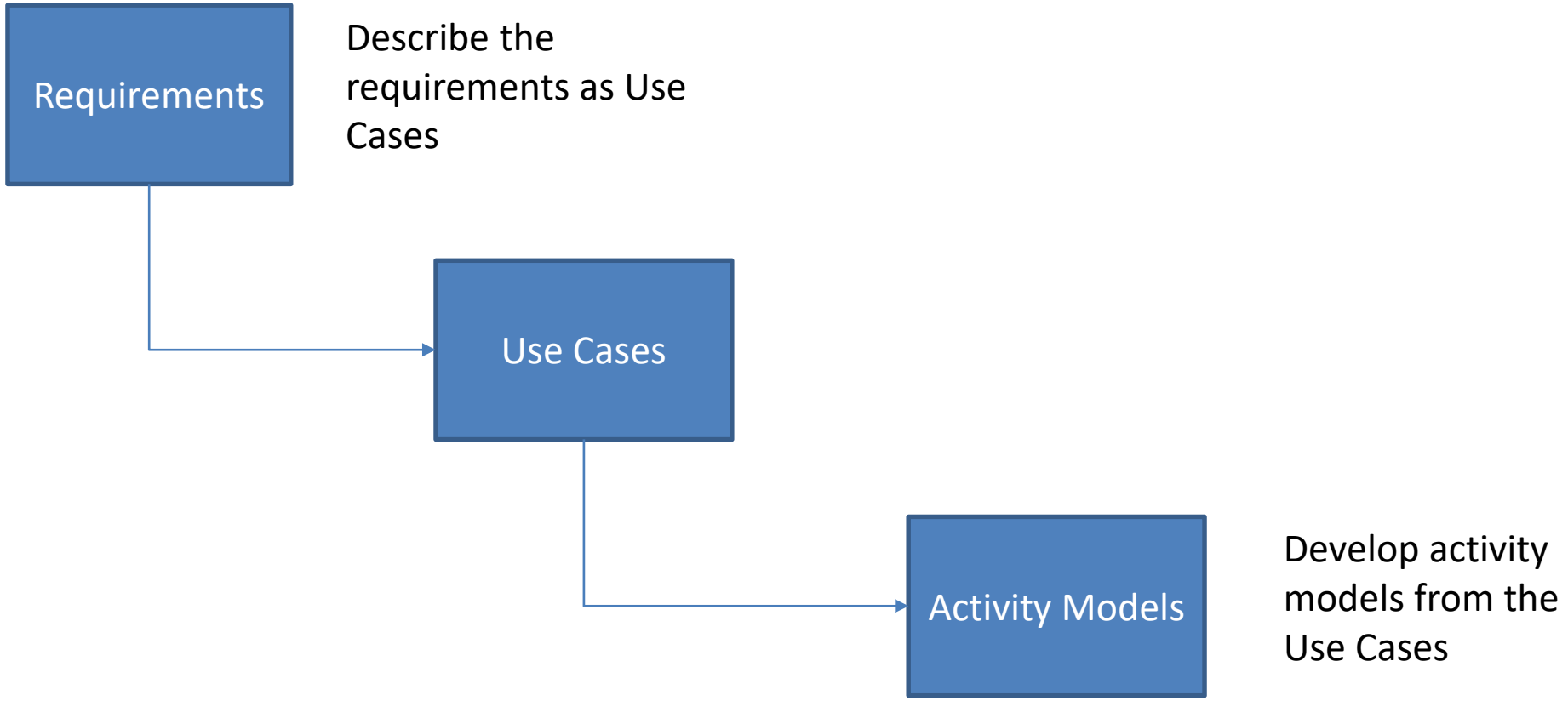
© Brian Gannon, August 2016

Modelling - Introduction

- Models are very powerful tools for communicating the analyst's understanding of requirements with the user
- Models are logical (independent of how they are implemented)
- Business process models describe the activities that collectively support a business process
- Use Cases are the primary driver for UML modelling: describes functions performed by users of a system
- Activity diagrams are used to model the behavior in a business process

© Brian Gannon, August 2016

Modelling - Introduction



© Brian Gannon, August 2016

UML Diagrams

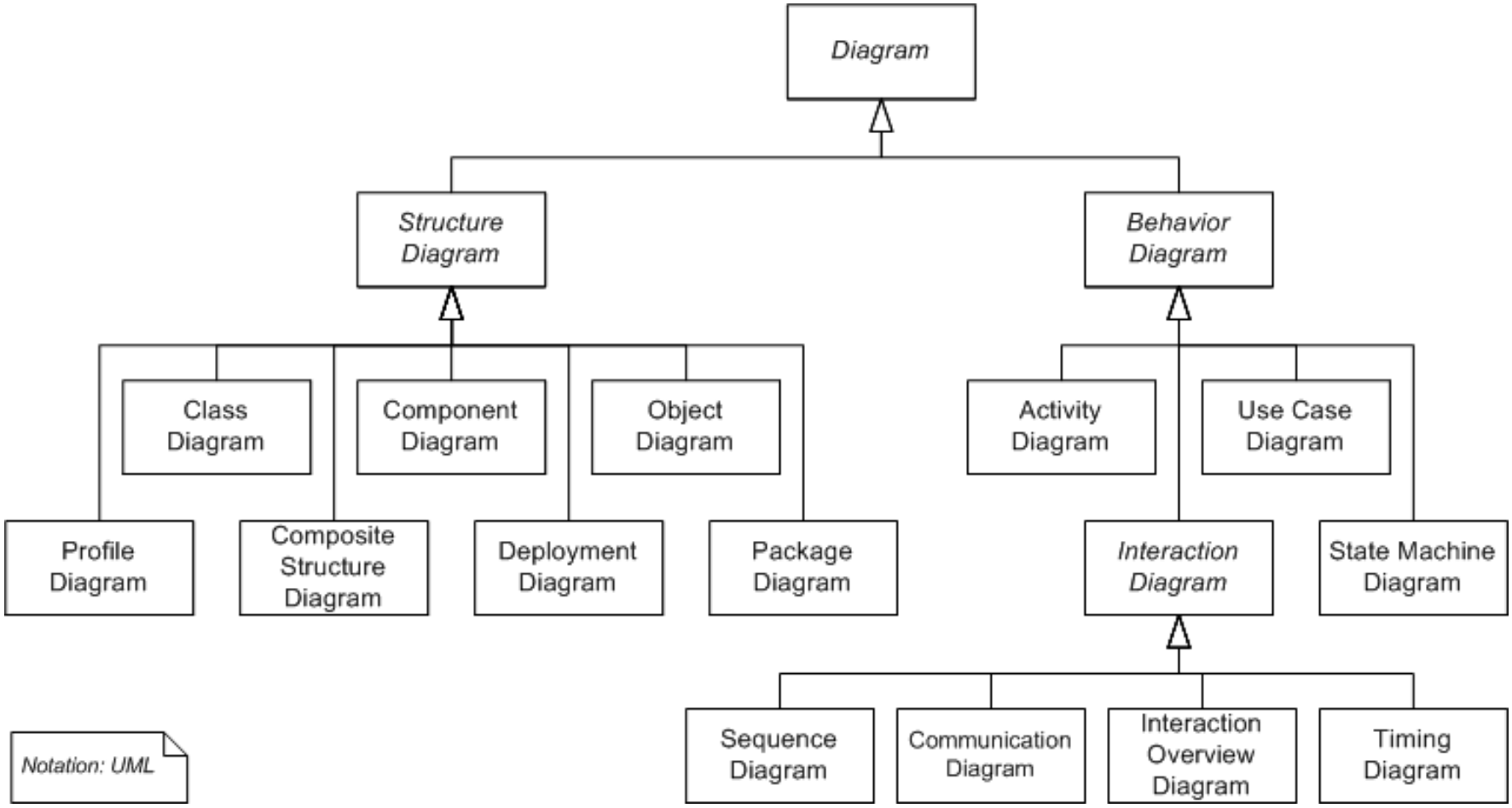
- UML diagrams represent two different views of a system model

    1. **Static (or Structural) view**: emphasises the static structure of the system using objects, attributes, operations and relationships. It includes class diagrams and composite structure diagrams.

    2. **Dynamic (or Behavioral) view**: emphasises the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects. This view includes sequence diagrams, activity diagrams and state machine diagrams.

© Brian Gannon, August 2016

Modelling - UML Diagram Hierarchy



© Brian Gannon, August 2016

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

Use Cases

- a use case describes functions performed by users of a system (basic functions, what the user can do, how the system responds)

- usually a diagram supported with a text documents (description)

- Each use case describes **one and only one function** – the discrete activities the users perform

© Brian Gannon, August 2016

Types of Use Cases

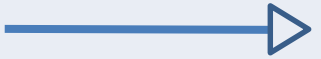
- a use case describes functions performed by users of a system (basic functions, what the user can do, how the system responds)
- usually a diagram supported with a text documents (description)
- Each use case describes **one and only one function** – the discrete activities the users perform

Purpose		Amount of information: Overview	Amount of information: Detail
	Essential	High-level **overview** of issues **essential** to understanding required functionality	**Detailed** description of issues **essential** to understanding required functionality
	Real	High-level **overview** of a specific set of steps performed on the **real** system once implemented	**Detailed** description of a specific set of steps performed on the **real** system once implemented

© Tegarden, Dennis, & Wixom, *Systems Analysis and Design with UML, 4th Edition*

© Brian Gannon, August 2016

Use Case Diagram Syntax

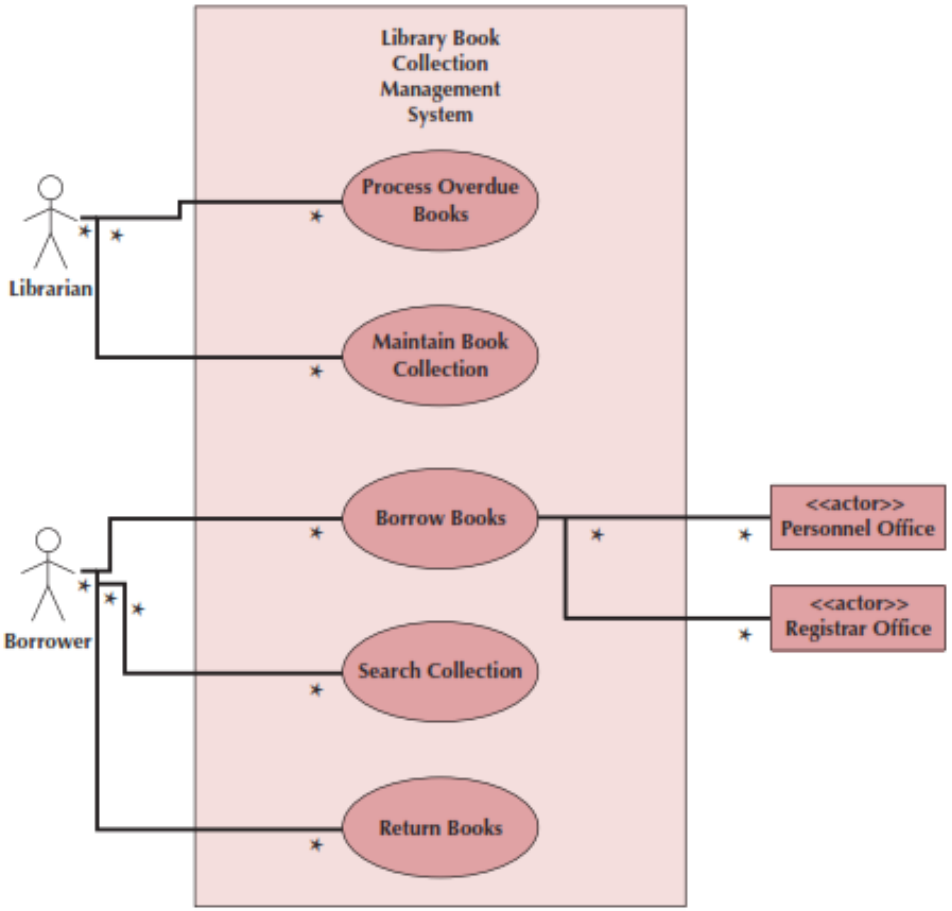
Element	Definition	Symbol
Actor	person or system that derives benefit from and is external to the subject	
Use Case	represents a major piece of system functionality	
Association Relationship		
Include Relationship		
Extend Relationship		
Generalisation Relationship		

© Brian Gannon, August 2016

Use Case Diagrams

- a use case describes functions performed by users of a system (basic functions, what the user can do, how the system responds)
- usually a diagram supported with a text documents (description)
- Each use case describes **one and only one function** – the discrete activities the users perform

Use Case Diagram: Library Management System



© Tegarden, Dennis, & Wixom, *Systems Analysis and Design with UML, 4th Edition*

Use Case Description

- **Overview:**
  - Name, ID Number, Type, Primary Actor, Brief Description, Importance Level, Stakeholder(s), Trigger(s)

- **Relationships**:
  - Association: communication between the use case and the actors
  - Extend: extends the functionality of a use case
  - Include: includes another use case
  - Generalisation: allows use cases to support inheritance

- **Flow of events:**
  - Normal flow: the usual set of activities
  - Sub-flows: decomposed normal flows to simplify the use-case
  - Alternate or exceptional flows: those not considered the norm

- **Optional characteristics:** (complexity, time, etc.)

© Brian Gannon, August 2016

Use Cases

- a use case describes functions that are performed by users of a system (basic functions, what the user can do, how the system responds)
- usually a diagram supported with a text documents (description)
- Each use case describes **one and only one function** – the discrete activities the users perform

Use Case Name: Borrow Books	ID: 2	Importance Level: High
Primary Actor: Borrower	Use Case Type: Detail, Essential	

Stakeholders and Interests:
Borrower - wants to check out books
Librarian - wants to ensure borrower only gets books deserved

Brief Description: This use case describes how books are checked out of the library.

Trigger: Borrower brings books to check out desk.
Type: External

Relationships:
Association: Borrower, Personnel Office, Registrar's Office
Include:
Extend:
Generalization:

Normal Flow of Events:
1. The Borrower brings books to the Librarian at the check out desk.
2. The Borrower provides Librarian their ID card.
3. The Librarian checks the validity of the ID Card.
   If the Borrower is a Student Borrower, Validate ID Card against Registrar's Database.
   If the Borrower is a Faculty/Staff Borrower, Validate ID Card against Personnel Database.
   If the Borrower is a Guest Borrower, Validate ID Card against Library's Guest Database.
4. The Librarian checks whether the Borrower has any overdue books and/or fines.
5. The Borrower checks out the books.

SubFlows:

Alternate/Exceptional Flows:
4a. The ID Card is invalid, the book request is rejected.
5a. The Borrower either has overdue books, fines, or both, the book request is rejected.

© Brian Gannon, August 2016

© Tegarden, Dennis, & Wixom, *Systems Analysis and Design with UML, 4th Edition*

Identifying Use Cases

- Review the requirements definition
- Identify the subject's boundaries
- Identify the primary actors and their goals
- Identify the business processes and major use-cases
- Carefully review the current set of use-cases
  - Split or combine some to create the right size
  - Identify additional use-cases

© Brian Gannon, August 2016

Creating Use Cases - Guidelines

1. Write In the form: Subject-Verb-Direct Object
2. Make sure initiator of a step is clear
3. Write from the point of view of an independent observer
4. Ensure consistent level of abstraction
5. Include a reasonable number of steps
6. Keep It Simple
7. Write repeating instructions after steps to be repeated
8. Ensure steps not too complicated/long & roughly same size as others
9. Identify exceptional work flows
10. Review and confirm
11. Iterate (all steps)

© Brian Gannon, August 2016

Creating Use Cases - Guidelines

1. Pick a high priority use-case and create an overview:
   - List the primary actor
   - Determine its type (overview or detail; essential or real)
   - List all stakeholders and their interests
   - Determine the level of importance of the use-case
   - Briefly describe the use-case
   - List what triggers the use-case
   - List its relationship to other use-cases

2. Fill in the steps of the normal flow of events required to complete the use-case

3. Conduct interactive review of models/diagrams with dev team and client before beginning structural and behavioral modeling (validation)

© Brian Gannon, August 2016

Rules For Validating A Use Case

- One and only one description for each use-case
- All objects in an activity diagram must be mentioned in use-case description
- Sequence of use-case description should match sequence in the activity diagram
- All actors listed in use-case description must be shown on use-case diagram (stakeholders do not all need to be shown – depends on local practice)
- All relationships in use-case description must be depicted on use-case diagram

© Brian Gannon, August 2016

Exercise

Draw a use case diagram and description

© Brian Gannon, August 2016

Identifying Main Use Cases and Associations

- Review the requirements definition
- Identify the subject's boundaries
- Identify the primary actors and their goals
- Identify the business processes and major use-cases
- Identify messages implying associations and sub-flows

© Brian Gannon, August 2016

Use Case Exercise – Requirements Definition

ISDSM Staff Fulfilment System

International Systems Development Staff Management (ISDSM) provide systems development staff e.g. Project Managers, Systems Designers, Computer Programmers to assist with new projects and major maintenance sub-projects. ISDSM negotiates with client companies to provide highly skilled temporary staff in specific job categories for a specified cost. When a client decides that it will need a temporary employee it issues a request against the contract it has previously negotiated with ISDSM. The staffing request will be processed by one of ISDSM's contract managers who will enter the contract number on the staffing request into the contract database. The type of staff required and anticipated cost will also be entered.

Using information from the database, the contract manager reviews the contract to determine whether the request is valid. A request is valid if there is an existing contract which lists the type of employee sought and the fee proposed is within the range previously negotiated. If the request is not valid the contract manager sends a letter to the client explaining the problem. If it is valid, the contract manager opens a staffing request.

The placement managers will scan the database for new staffing requests and free staff who may be able to fulfil the requests. If a placement manager finds a match they will place a reservation against both the prospective employee and the request while arrangements are made. If arrangements are made they will complete the records concerning the expected length of the engagement involving the triggering of messages to the client and the prospective temporary employee. The contract manager will periodically review outstanding requests and liaise with the client.

© Brian Gannon, August 2016

Use Case Exercise – Identify System Boundaries

ISDSM Staff Fulfilment System

International Systems Development Staff Management (ISDSM) provide systems development staff e.g. Project Managers, Systems Designers, Computer Programmers to assist with new projects and major maintenance sub-projects. ISDSM negotiates with client companies to provide highly skilled temporary staff in specific job categories for a specified cost. When a client decides that it will need a temporary employee it issues a request against the contract it has previously negotiated with ISDSM. The staffing request will be processed by one of ISDSM's contract managers who will enter the contract number on the staffing request into the contract database. The type of staff required and anticipated cost will also be entered.

Using information from the database, the contract manager reviews the contract to determine whether the request is valid. A request is valid if there is an existing contract which lists the type of employee sought and the fee proposed is within the range previously negotiated. If the request is not valid the contract manager sends a letter to the client explaining the problem. If it is valid, the contract manager opens a staffing request.

The placement managers will scan the database for new staffing requests and free staff who may be able to fulfil the requests. If a placement manager finds a match they will place a reservation against both the prospective employee and the request while arrangements are made. If arrangements are made they will complete the records concerning the expected length of the engagement involving the triggering of messages to the client and the prospective temporary employee. The contract manager will periodically review outstanding requests and liaise with the client.

© Brian Gannon, August 2016

Identify Primary Actors

ISDSM Staff Fulfilment System

International Systems Development Staff Management (ISDSM) provide systems development staff e.g. Project Managers, Systems Designers, Computer Programmers to assist with new projects and major maintenance sub-projects. ISDSM negotiates with client companies to provide highly skilled temporary staff in specific job categories for a specified cost. When a client decides that it will need a temporary employee it issues a request against the contract it has previously negotiated with ISDSM. The staffing request will be processed by one of ISDSM's contract managers who will enter the contract number on the staffing request into the contract database. The type of staff required and anticipated cost will also be entered.

Using information from the database, the contract manager reviews the contract to determine whether the request is valid. A request is valid if there is an existing contract which lists the type of employee sought and the fee proposed is within the range previously negotiated. If the request is not valid the contract manager sends a letter to the client explaining the problem. If it is valid, the contract manager opens a staffing request.

The placement managers will scan the database for new staffing requests and free staff who may be able to fulfil the requests. If a placement manager finds a match they will place a reservation against both the prospective employee and the request while arrangements are made. If arrangements are made they will complete the records concerning the expected length of the engagement involving the triggering of messages to the client and the prospective temporary employee. The contract manager will periodically review outstanding requests and liaise with the client.

© Brian Gannon, August 2016

Identify Primary Actors and their Goals and Constraints

ISDSM Staff Fulfilment System

International Systems Development Staff Management (ISDSM) provide systems development staff e.g. Project Managers, Systems Designers, Computer Programmers to assist with new projects and major maintenance sub-projects. ISDSM negotiates with client companies to **provide highly skilled temporary staff in specific job categories for a specified cost.** When a client decides that it will need a temporary employee it issues a request against the contract it has previously negotiated with ISDSM. The staffing request will be processed by one of ISDSM's contract managers who will enter the contract number on the staffing request into the contract database. The type of staff required and anticipated cost will also be entered.

Using information from the database, the contract manager reviews the contract to determine whether the request is valid. A request is valid if there is an existing contract which lists the type of employee sought and the **fee proposed is within the range previously negotiated.** If the request is not valid the contract manager sends a letter to the client explaining the problem. If it is valid, the contract manager opens a staffing request.

The placement managers will scan the database for new staffing requests and free staff who may be able to fulfil the requests. If a placement manager finds a match they will place a reservation against both the prospective employee and the request while arrangements are made. If arrangements are made they will complete the records concerning the expected length of the engagement involving the triggering of messages to the client and the prospective temporary employee. The contract manager will periodically review outstanding requests and liaise with the client.

© Brian Gannon, August 2016

Identify Business Processes and Main Use Cases

ISDSM Staff Fulfilment System

International Systems Development Staff Management (ISDSM) provide systems development staff e.g. Project Managers, Systems Designers, Computer Programmers to assist with new projects and major maintenance sub-projects. ISDSM negotiates with client companies to **provide highly skilled temporary staff in specific job categories for a specified cost**. When a client decides that it will need a temporary employee **it issues a request (1)** against the contract it has previously negotiated with ISDSM. **The staffing request will be processed (2)** by one of ISDSM's contract managers who will enter the contract number on the staffing request into the contract database. The type of staff required and anticipated cost will also be entered.

Using information from the database, the contract manager reviews the contract to determine whether the request is valid. A request is valid if there is an existing contract which lists the type of employee sought and the **fee proposed is within the range previously negotiated.** If the request is not valid the contract manager sends a letter to the client explaining the problem. If it is valid, the contract manager opens a staffing request.

The placement managers will scan the database for new staffing requests and free staff who may be able to **fulfil the requests (3)**. If a placement manager finds a match they will place a reservation against both the prospective employee and the request while arrangements are made. If arrangements are made they will complete the records concerning the expected length of the engagement involving the triggering of messages to the client and the prospective temporary employee. The contract manager will periodically review outstanding requests and liaise with the client.

© Brian Gannon, August 2016

Identify Subflows/Associations

ISDSM Staff Fulfilment System

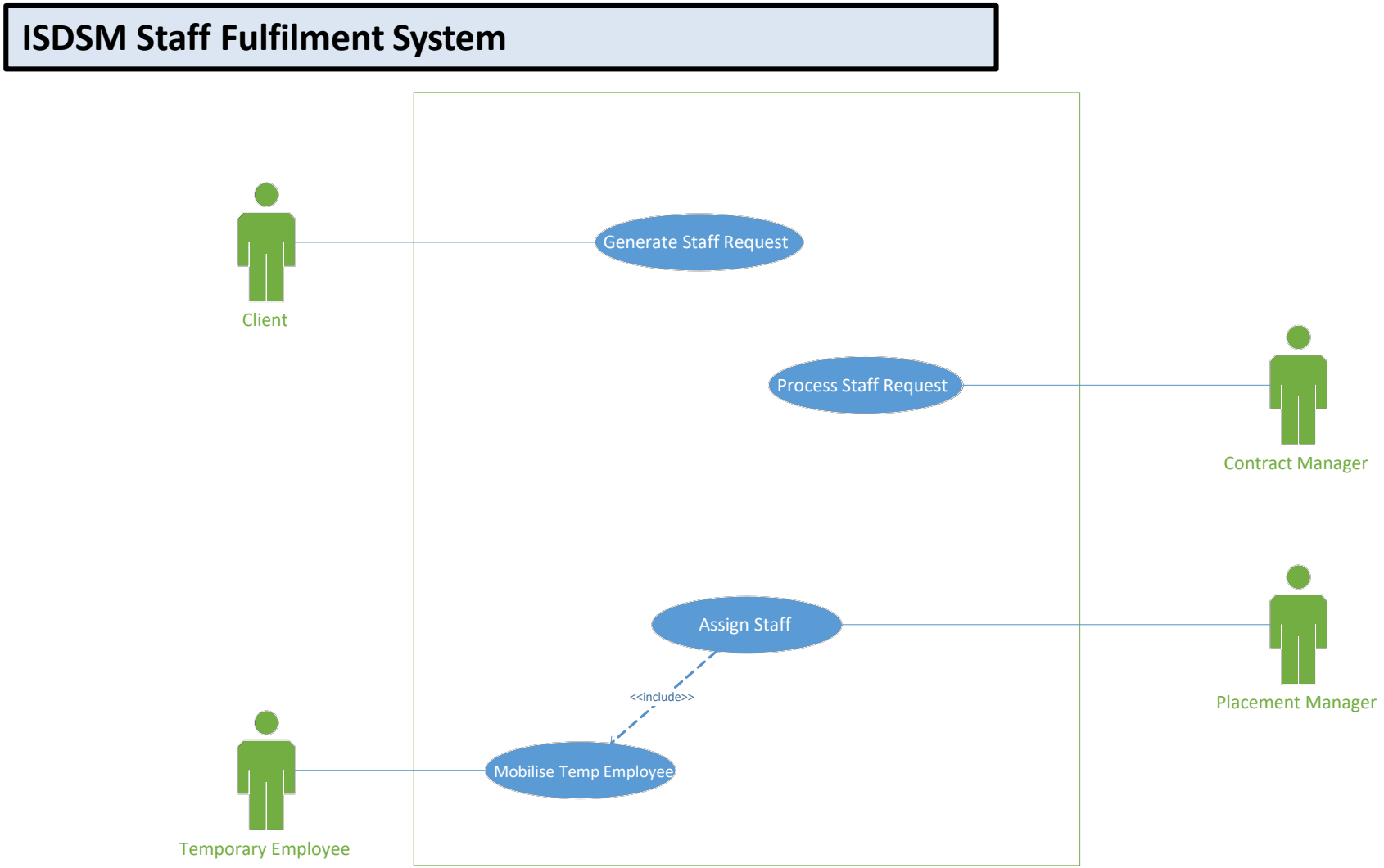
International Systems Development Staff Management (ISDSM) provide systems development staff e.g. Project Managers, Systems Designers, Computer Programmers to assist with new projects and major maintenance sub-projects. ISDSM negotiates with client companies to **provide highly skilled temporary staff in specific job categories for a specified cost**. When a client **decides that it will need a temporary employee (1.1) it issues a request (1)** against the contract it has previously negotiated with ISDSM. **The staffing request will be processed (2)** by one of ISDSM's contract managers who will **enter the contract number on the staffing request into the contract database. The type of staff required and anticipated cost will also be entered. (2.1)**

Using information from the database, the contract manager **reviews the contract to determine whether the request is valid (2.2)**. A request is valid if there is an existing contract which lists the type of employee sought and the **fee proposed is within the range previously negotiated.** If the request is not valid the **contract manager sends a letter to the client (2.3)** explaining the problem. If it is valid, the contract manager **opens a staffing request (2.3)**.

The placement managers will **scan the database for new staffing requests (3.1)** and free staff who may be able to **fulfil the requests (3)**. If a placement manager finds a match they will **place a reservation against both the prospective employee and the request (3.2)** while arrangements are made. If arrangements are made they will **complete the records (3.3)** concerning the expected length of the engagement involving the triggering of messages to the client and the prospective temporary employee. The contract manager will **periodically review outstanding requests (2.4)** and **liaise with the client (2.4)**.

© Brian Gannon, August 2016

Use Case Diagram



© Brian Gannon, August 2016

Use Case Description Exercise

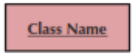
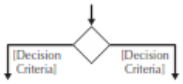
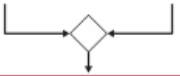
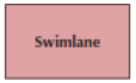
Use Case Name: **Fulfil Staff**	ID:	Importance Level:
Primary Actor:	Use Case Type:	
Stakeholders and Interests:		
Brief Description:		
Trigger: Type:		
Relationships: Association: Include: Extend: Generalization:		
Normal Flow of Events: 1.		
Subflows: 1. 2. 3.		
Alternate/Exceptional Flows:		

© Brian Gannon, August 2016

Activity Diagrams

- used to model business processes (how a business operates)
- developed from use cases
- used to illustrate the movement of objects (data) between activities
- abstract – they describe processes in general
- can be used for any type of process

Activity Diagrams - Syntax

An action: ■ Is a simple, nondecomposable piece of behavior. ■ Is labeled by its name.	
An activity: ■ Is used to represent a set of actions. ■ Is labeled by its name.	
An object node: ■ Is used to represent an object that is connected to a set of object flows. ■ Is labeled by its class name.	
A control flow: ■ Shows the sequence of execution.	
An object flow: ■ Shows the flow of an object from one activity (or action) to another activity (or action).	
An initial node: ■ Portrays the beginning of a set of actions or activities.	
A final-activity node: ■ Is used to stop all control flows and object flows in an activity (or action).	
A final-flow node: ■ Is used to stop a specific control flow or object flow.	
A decision node: ■ Is used to represent a test condition to ensure that the control flow or object flow only goes down one path. ■ Is labeled with the decision criteria to continue down the specific path.	
A merge node: ■ Is used to bring back together different decision paths that were created using a decision node.	
A fork node: Is used to split behavior into a set of parallel or concurrent flows of activities (or actions)	
A join node: Is used to bring back together a set of parallel or concurrent flows of activities (or actions)	
A swimlane: Is used to break up an activity diagram into rows and columns to assign the individual activities (or actions) to the individuals or objects that are responsible for executing the activity (or action) Is labeled with the name of the individual or object responsible	

© Tegarden, Dennis, & Wixom, *Systems Analysis and Design with UML, 4th Edition*

Activity Diagrams - Syntax

- Actions & Activities
  - Something performed for some specific business reason
  - Named with a verb and a noun (e.g. Get Patient Information)
  - Activities can be further sub-divided; actions cannot
- Object Nodes: represent the flow of information from one activity to another
- Control Flows: model execution paths
- Object Flows: model the flow of objects
- Control Nodes: 7 types

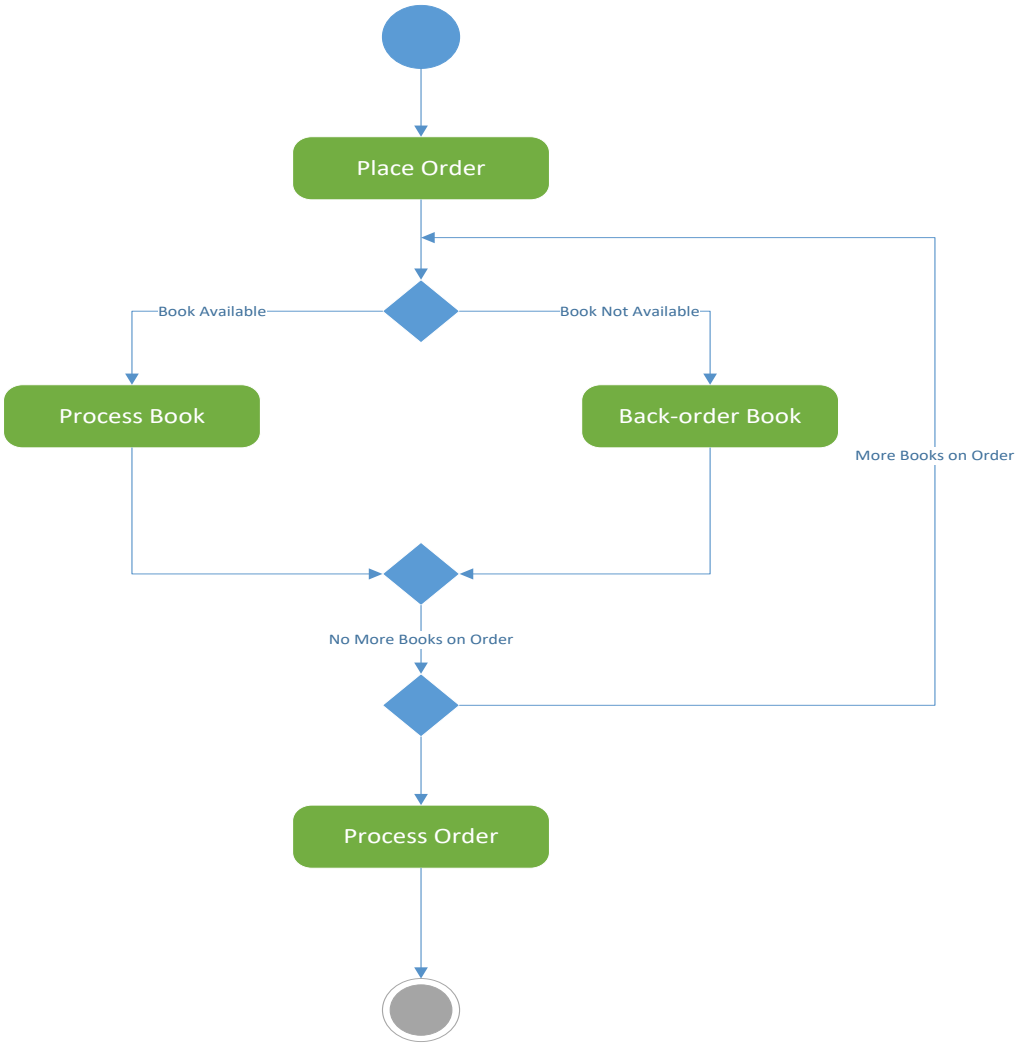
© Brian Gannon, August 2016

Activity Diagrams – Control Nodes

- Initial node: the beginning of the set of actions/activities
- Final-activity node: stops all actions/activities
- Final-flow node: stops one execution path but allows others to continue
- Decision node: test to determine which path to use to continue (guard condition)
- Merge node: rejoins mutually exclusive execution paths
- Fork node: separates a single execution path into one or more parallel paths
- Join node: rejoins parallel execution paths

© Brian Gannon, August 2016

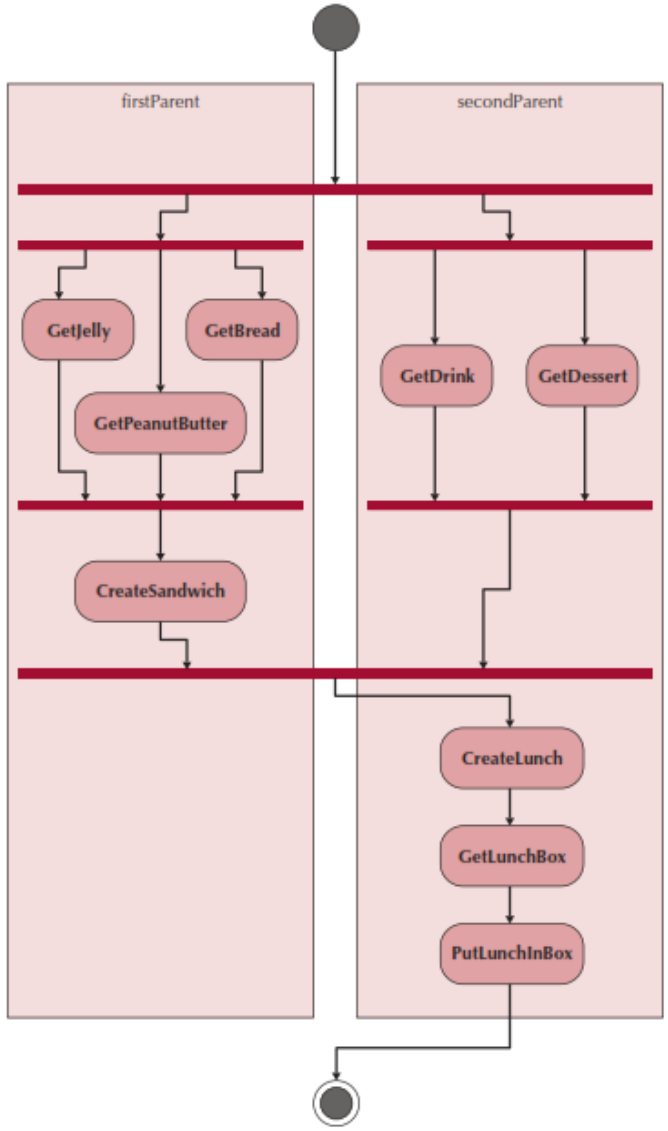
Activity Diagram – Example



© Brian Gannon, August 2016

Activity Diagrams – Swim Lanes

- Used to assign responsibility to objects or individuals who actually perform the activity
- Represents a separation of roles among objects
- Can be drawn horizontally or vertically



© Tegarden, Dennis, & Wixom, *Systems Analysis and Design with UML, 4th Edition*

Activity Diagrams - Guidelines

- Choose a business process
  - Review the requirements definition and use-case diagram
  - Review other documentation
- Identify the **set of activities** used in the business process
- Identify **control flows and nodes**
- Identify the **object flows and nodes**
- Lay out and draw the diagram (minimise crossing lines)

© Brian Gannon, August 2016

Exercise

Draw an activity diagram

© Brian Gannon, August 2016

Activity Diagram - Exercise

Use Case Name:	Make Appointment	ID: 2	Importance Level: High
Primary Actor:	Patient	Use Case Type:	Detail, Essential

Stakeholders and Interests:
Patient – wants to make, change or cancel and appointment
Doctor – wants to ensure patients' needs are met

Brief Description: This use case describes how a person can make, change or cancel and appointment

Trigger: Patient Calls and asks for a new appointment or asks to cancel or change an existing appointment
Type: External

Relationships:
- Association: Patient
- Include: Make Payment
- Extend: Create new patient
- Generalization:

Normal Flow of Events:
1. Patient contacts the office
2. Patient provides receptionist with her name and address
3. Receptionist validates that the patient is registered and is enrolled in patient database
4. Receptionist executes Make Payment Arrangements use case
5. Receptionist asks Patient if she would like to make new or change or cancel existing appointment
   a. If Patient wants new appointment, then the New Appointment sub-flow (S-1) is performed
   b. If Patient wants to change appointment, then the Change Appointment sub-flow (S-2) is performed
   c. If Patient wants to cancel appointment, then the Cancel Appointment sub-flow (S-3) is performed
6. Receptionist provides results of transaction to the Patient

Subflows:
1. New Appointment
   1. Receptionist asks Patient for suitable appointment times
   2. Receptionist matches Patient's desired times with available dates and times and schedules a new appointment
2. Cancel Appointment
   1. Receptionist asks Patient for old appointment time
   2. Receptionist finds current appointment and cancels it
3. Change Appointment
   1. Receptionist performs Cancel Appointment (S-2) subflow
   2. Receptionist performs New Appointment (S-1) subflow

Alternate/Exceptional Flows:
- 3a. Receptionist performs Create New Patient use case
- S-1, 2a1: Receptionist proposes alternative appointment times based on availability of doctor showing in schedule
- S-1, 2a2: Patient chooses one of the proposed times or declines to make an appointment

© Brian Gannon, August 2016

References

Tegarden, Dennis, & Wixom, *Systems Analysis and Design with UML, 4th Edition*

Dr. BRIAN GANNON

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