

# COMP3004 Midterm Notes

*William Findlay*

*February 17, 2019*

# Contents

<b>1</b>	<b>Software Engineering</b>	<b>1</b>
<b>2</b>	<b>Build Models</b>	<b>1</b>
2.1	Functional Model (Elicitation) . . . . .	1
2.1.1	Use Cases (Tables and Diagrams) . . . . .	1
2.1.2	FURPS+ Requirements (Tables) . . . . .	3
2.2	Dynamic Model (Analysis) . . . . .	4
2.2.1	State Machines . . . . .	4
2.2.2	Sequence Diagrams . . . . .	4
2.2.3	Activity Diagrams . . . . .	4
2.3	Object Model (Analysis) . . . . .	4
2.3.1	Class Diagrams . . . . .	4
2.3.2	Data Dictionaries . . . . .	5
2.4	Traceability . . . . .	5
<b>3</b>	<b>Software Development Life Cycle</b>	<b>5</b>
<b>4</b>	<b>Requirements Elicitation</b>	<b>6</b>
<b>5</b>	<b>Analysis</b>	<b>6</b>
<b>6</b>	<b>High Level System Design</b>	<b>6</b>

## List of Figures

2.1	Components of use case diagrams and tables. . . . .	1
2.2	Example high level use case diagram. . . . .	2
2.3	Example detailed use case diagram. . . . .	2
2.4	Inheritance, composition, and aggregation in UML class diagrams. . . . .	5

## List of Tables

2.1	An example use case table for a high level use case. . . . .	3
2.2	An example use case table for an extend use case. . . . .	3
2.3	An example functional requirements table. . . . .	4
2.4	An example non-functional requirements table. . . . .	4

## List of Listings

# 1 Software Engineering

- what is it?
  - requirements analysis
  - building a *software system*
- why is it necessary?
  - systems get huge and difficult to manage
  - we need a plan
  - *reliability*
  - *modifiability*

## 2 Build Models

- what is a model?
  - representation of how to build system
  - get a better idea of how to do it
  - clarify requirements

### 2.1 Functional Model (Elicitation)

- use case diagrams
- use case tables
- FR, NFR tables

#### 2.1.1 Use Cases (Tables and Diagrams)

- see Figure 2.1 for components of use case diagrams and tables
- see Figure 2.2 for an example high level use case diagram
- see Figure 2.3 for an example detailed use case diagram
- see Table 2.1 and Table 2.2 for example use case tables

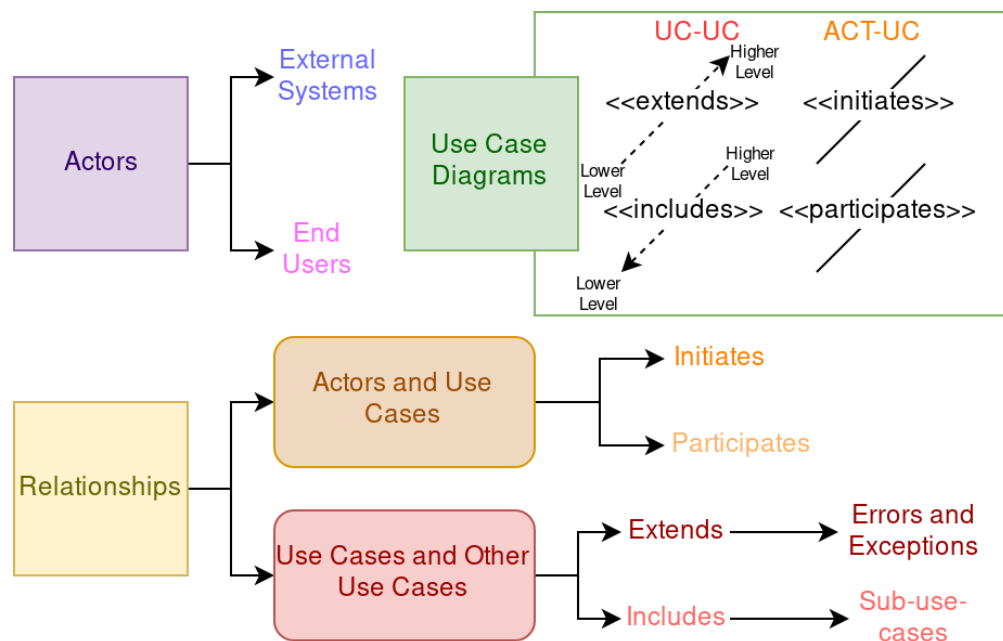


Figure 2.1: Components of use case diagrams and tables.



Figure 2.2: Example high level use case diagram.

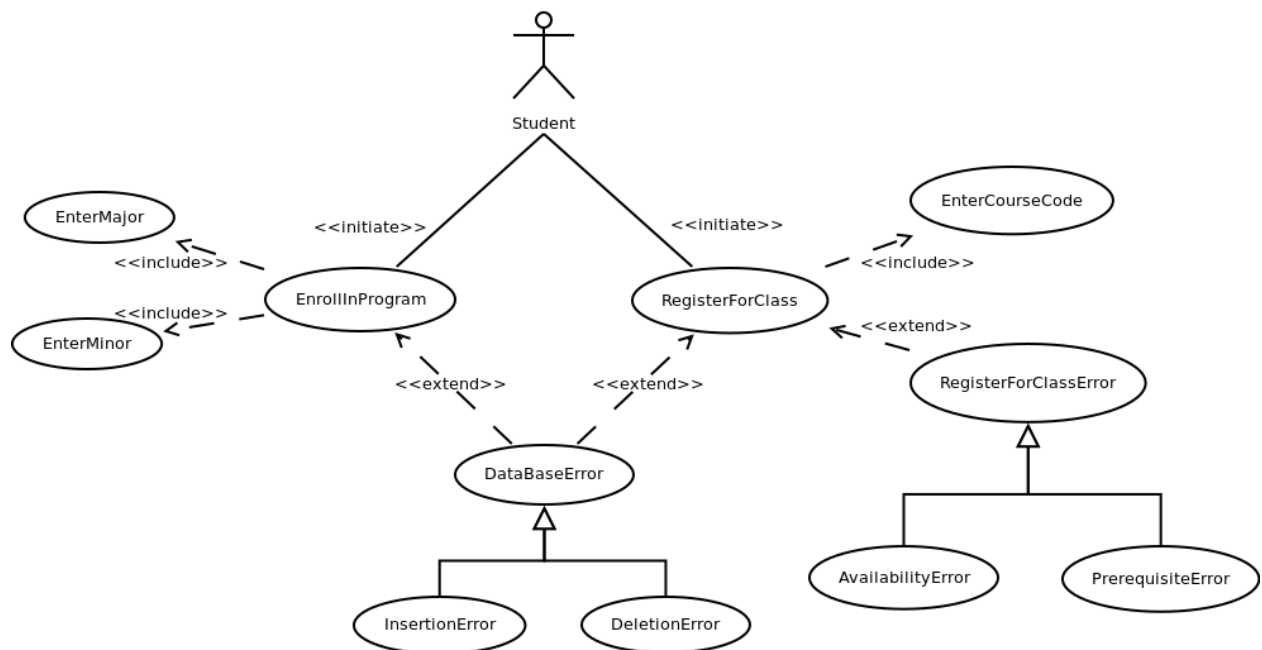


Figure 2.3: Example detailed use case diagram.

**Table 2.1:** An example use case table for a high level use case.

Number	UC-01
Name	RegisterForClass
Participating Actors	Initiated by: Student Participated in by: External Database
Flow of Events	<ol style="list-style-type: none"> <li>1. Student selects the option to register for a class</li> <li>2. Student enters the desired course code (include use case EnterCourseCode)</li> <li>3. System fetches information for the course from the database</li> <li>4. System checks to see if student is available for the course's time slot</li> <li>5. System checks to see if student meets prerequisites</li> <li>6. System registers student for the course in the database</li> <li>7. System notifies student that they have been registered successfully</li> </ol>
Entry Condition	● Student is logged in
Exit Condition	● Student is registered for the course in the database
Quality Requirements	<ul style="list-style-type: none"> <li>● Student must be notified once they are registered</li> <li>● Student cannot register for two courses in the same time slot</li> </ul>
Traceability	FR-03, NFR-21, NFR-23

**Table 2.2:** An example use case table for an extend use case.

Number	UC-07
Name	RegisterForClassError
Participating Actors	Student, External Database
Flow of Events	1. System notifies student that there was an error registering for
Entry Condition	<ul style="list-style-type: none"> <li>● This use case extends RegisterForClass</li> <li>● Initiated when the system detects an error registering for the desired course</li> </ul>
Exit Condition	● The class registration is aborted
Quality Requirements	● Student must be notified when there is an error
Traceability	NFR-22

### 2.1.2 FURPS+ Requirements (Tables)

**F**unctional

**U**sability

**R**eliability

**P**erformance

**S**upportability

+ Operation, Interface, Implementation, Packaging, Legal

- see Table 2.3 for a functional requirements table
- see Table 2.4 for a non-functional requirements table

**Table 2.3:** An example functional requirements table.

Number	Functional Requirement
FR-01	Student can register for classes.
FR-02	Student can enroll in a program.
FR-03	Staff and students can sign up.
FR-04	Staff and students can log in.
...	...

**Table 2.4:** An example non-functional requirements table.

Number	Category	Non-Functional Requirement
NFR-01	Usability	No operation within the software should take more than three context menus to complete
NFR-02	Reliability	The software should be able to recover all data in the event of a system crash
NFR-03	Performance	No UI operation should take more than 1 second to provide feedback at least 95% of the time
NFR-04	Supportability	The system should be extensible to support GNU/Linux, MacOS, and Windows
NFR-05	Operation	Only staff should be able to execute management operations in the system
NFR-06	Interface	The UI should be professional and consistent with commercially available UIs
NFR-07	Implementation	Student profiles should contain a name, an age, and a student number.
NFR-08	Packaging	The system should be able to installed and run with a single command.
NFR-09	Legal	Students must be over the age of 18 or have parent permission to enrol, as required by local laws.

## 2.2 Dynamic Model (Analysis)

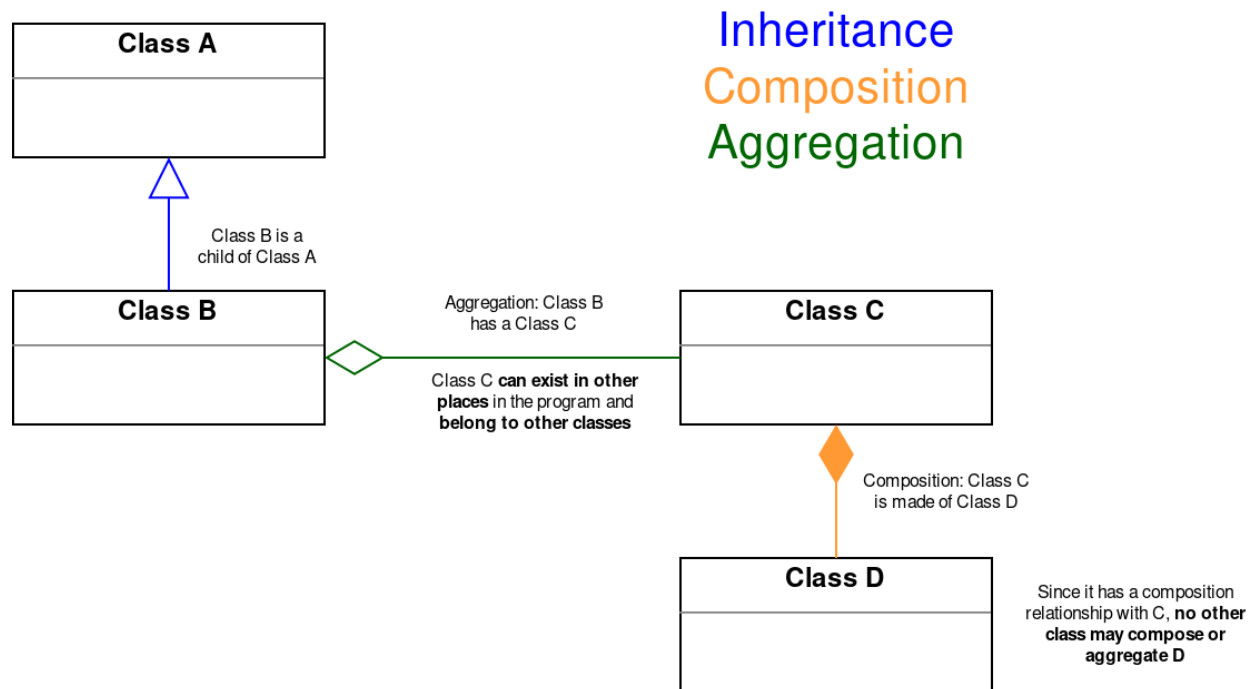
### 2.2.1 State Machines

### 2.2.2 Sequence Diagrams

### 2.2.3 Activity Diagrams

## 2.3 Object Model (Analysis)

### 2.3.1 Class Diagrams



**Figure 2.4:** Inheritance, composition, and aggregation in UML class diagrams.

### 2.3.2 Data Dictionaries

## 2.4 Traceability

- required changes?
  - > traceability lets us figure out *what parts are affected*
- numbers on all table rows
  - > FR-01, ...
  - > NFR-01, ...
  - > UC-01, ...

## 3 Software Development Life Cycle

1. Requirements Elicitation
2. Analysis

---

Client Knowledge Disappears

3. High Level System Design
4. Detailed Object Design
5. Implementation

---

Client Knowledge Reappears

6. Testing
7. Deployment and Maintenance



## 4 Requirements Elicitation

- what does the client want?
- requirements (FURPS+)
  - functional
    - what do the actors do?
  - non-functional
    - constraints
    - quality requirements
- scenarios, use cases
- work products
  - functional model
    - FR, NFR
    - use case diagrams

## 5 Analysis

- work products
  - object model
    - class diagrams
  - dynamic model
    - sequence diagrams
    - state machine diagrams
    - activity diagrams

## 6 High Level System Design