Modelling

Madiba Hudson-Quansah

Contents

Chapter 1

Principles that guide practice

Chapter 2

Unified Modelling Language (UML)

2.1 Introduction

Definition 2.1.1: UML

A standard graphical language for modelling object-oriented software systems.

UML can be used to model the following:

Class Diagrams - Describe classes and their relationships.

Interaction Diagrams - Show the behaviour of systems in terms of how they interact with each other.

State Diarams and Activity Diagrams - Show how systems behave internally

Component and Deployment Diagrams - Show how the various components of systems are arranged logically and physically.

2.2 Class Diagrams

The main symbols seen in class diagrams are:

Class - Represents the type of data, represented by a rectangle with three compartments.

Association - Represents linkage between instances of classes, represented by a line connecting the classes.

Attribute - Simple data found in classes and their instances, represented by a name and a type.

Operation - Represent the functions performed by classes, and their instances, represented by a name and a type.

Generalization - Groups classes into inheritance hierarchies, represented by a line with a triangle pointing to the superclass.

2.2.1 Class

Class Name - The name of the class.

Attributes - The data that the class holds, with format `visibility attributeName: type`. Where the available visibilities are:

`+` - Public

`-` - Private

`#` - Protected

`~` - Package

Operations - The functions that the class can performed, with signature `operationName(parameters): returnType` .

2.2.2 Associations and Multiplicity

Show that two classes are related to each other, with symbols indicating the multiplicity of the relationship at the end of the line. With the following symbols:

0..1 - Zero or one

0..* / * - Zero or more

n - Exactly n

$m..n$ - Between m and n

0, $m..n$ - Zero or between m and n

There are three classes of multiplicities:

Many-to-Many - Both classes can have many instances of the other, for example a student can take many courses and a course can be taken by many students.

Many-to-One / One-to-Many - One class can have many instances of the other, for example a branch can have many workers.

One-to-One - One class can have only one instance of the other, for example a person can have only one passport.

It is possible for an association to connect a class to itself, this is called a **self-association / reflexive association**. For example, a person can be married to another person.

Associations are bi-directional by default, but can be made one way by adding an arrowhead to the line.

2.2.3 Generalization

Generalization is used to show that one class is a superclass of another class. The superclass is the parent class, and the subclass is the child class. The subclass inherits the attributes and operations of the superclass. The discriminator is a label that describes the criteria used in the specialization