Introduction

Design Patterns and UML class diagrams

Linda Marshall

Department of Computer Science
University of Pretoria

24 July 2015

Overview

Definition 1

Patterns identify and specify abstractions that are above the level of single classes and instances or of components.

Gamma et al (1995) [Gang of Four (GoF)]

Definition 2

Design Patterns constitute a set of rules describing how to accomplish certain tasks in the realm of software development.

Pree (1995)

Definition 3

Design Patterns focus more on reuse of recurring architectural design themes, while frameworks focus on detail design and implementation.

Coplien and Schmidt (1995)

Definition 4

A pattern addresses a recurring design problem that arises in specific design situations and presents a solution to it.

Buschmann et al (1996)

Definition 5

Design Patterns are recurring solutions to design problems you see over and over.

The Smalltalk Companion (1998)

Definition 6

Experienced OO developers build up a repertoire of general principles and idiomatic solutions that guide them in the creation of software. These may be called patterns.

Craig Larman(2006)

Definition 7

Design Patterns are programming tools to improve code to be:

- easier to implement, and
- easier to maintain.
- are good answers to common and specialised problems.
- define a common (programming language independent) programming model that standardise common programming tasks into recognisable forms, giving your projects better cohesiveness.

CG Lasater (2007)

When design patterns are applied we achieve:

- Improved maintainability of code
- Improved adaptability of code
- Improved reliability of code
- Programmers who are more effective in their work.

There are 23 classic patterns categorised as:

- Creational
- Behavioural
- Structural

Each design pattern is further categorised by the strategy used in the implementation. The two implementation strategies are *Delegation* and *Inheritance*. These strategies are further refined in terms of the interaction between *classes* or the interaction between *objects*.

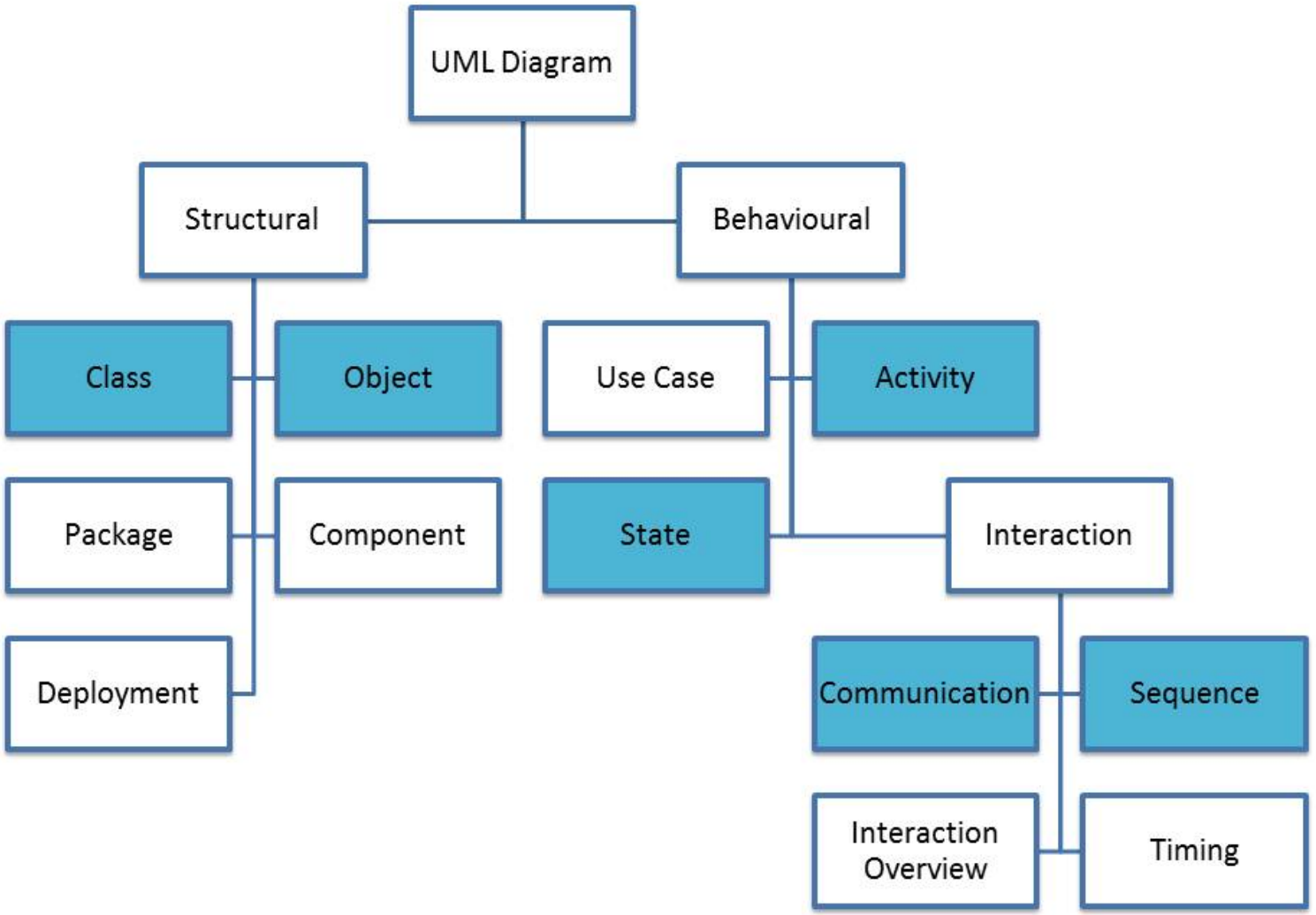
Unified Modelling Language (UML) is a standard notation for the modelling of real-world objects as a first step in developing an object-oriented design methodology.

`http://searchsoftwarequality.techtarget.com/definition/Unified-Modeling-Language`

UML *unifies* three OO methodologies:

- OMT - 1991 - James Rumbaugh
- OOA and OOD - early 1990's - while Grady Booch worked at Rationale
- OOSE - 1992 - Ivor Jacobson

The ideas of these "Three Amigos" and others, under the sponsorship of Rationale, lead to UML in 1995. UML 2 adapted by OMG in 2005.


UML Diagram
Structural
Behavioural
Class
Object
Use Case
Activity
Package
Component
State
Interaction
Deployment
Communication
Sequence
Interaction Overview
Timing

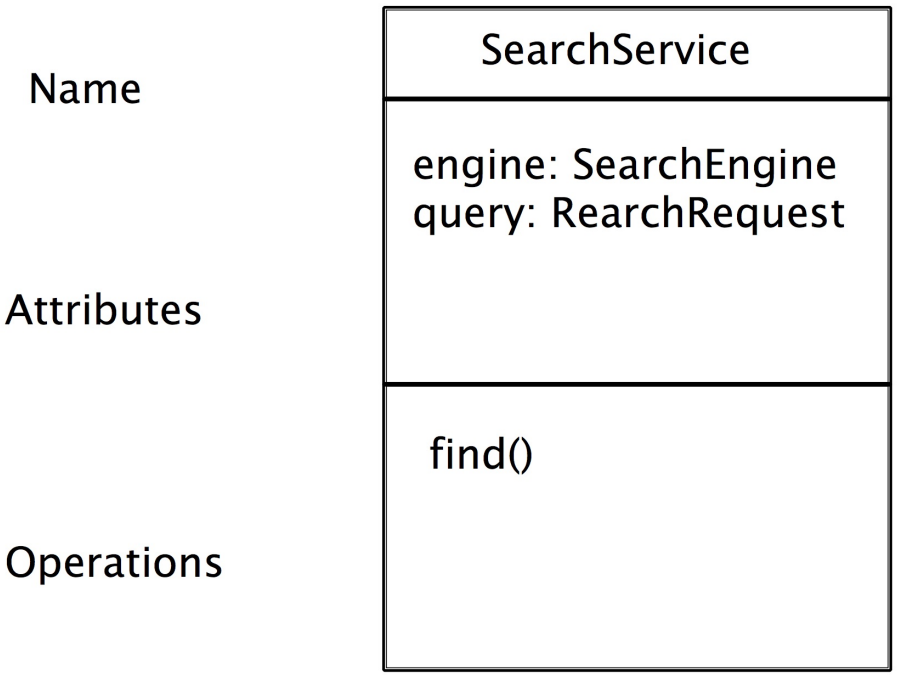
Class diagram is UML structure diagram which shows structure of the designed system at the level of classes and interfaces, shows their features, constraints and relationships - associations, generalisations, dependencies, etc.

A class is a classifier which describes a set of objects that share the same:

- features (or members)
- constraints
- semantics (meaning)

Features

Features of a class are **attributes** and **operations**.

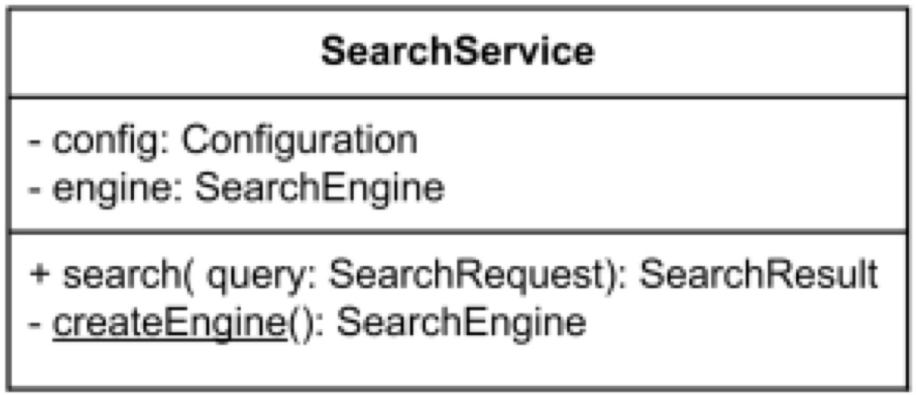


Features cont.

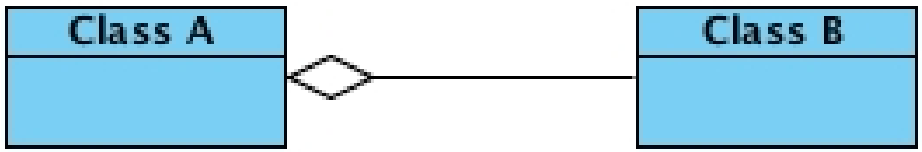
+ (public), # (protected) and − (private) are used to specify the visibility of the respective features.

Always list features in order from public to private in each of the sections of the class.

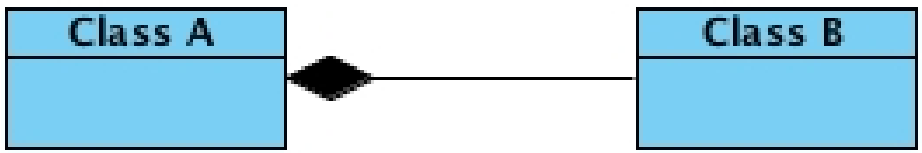
Static members are underlined.



Association



Aggregation



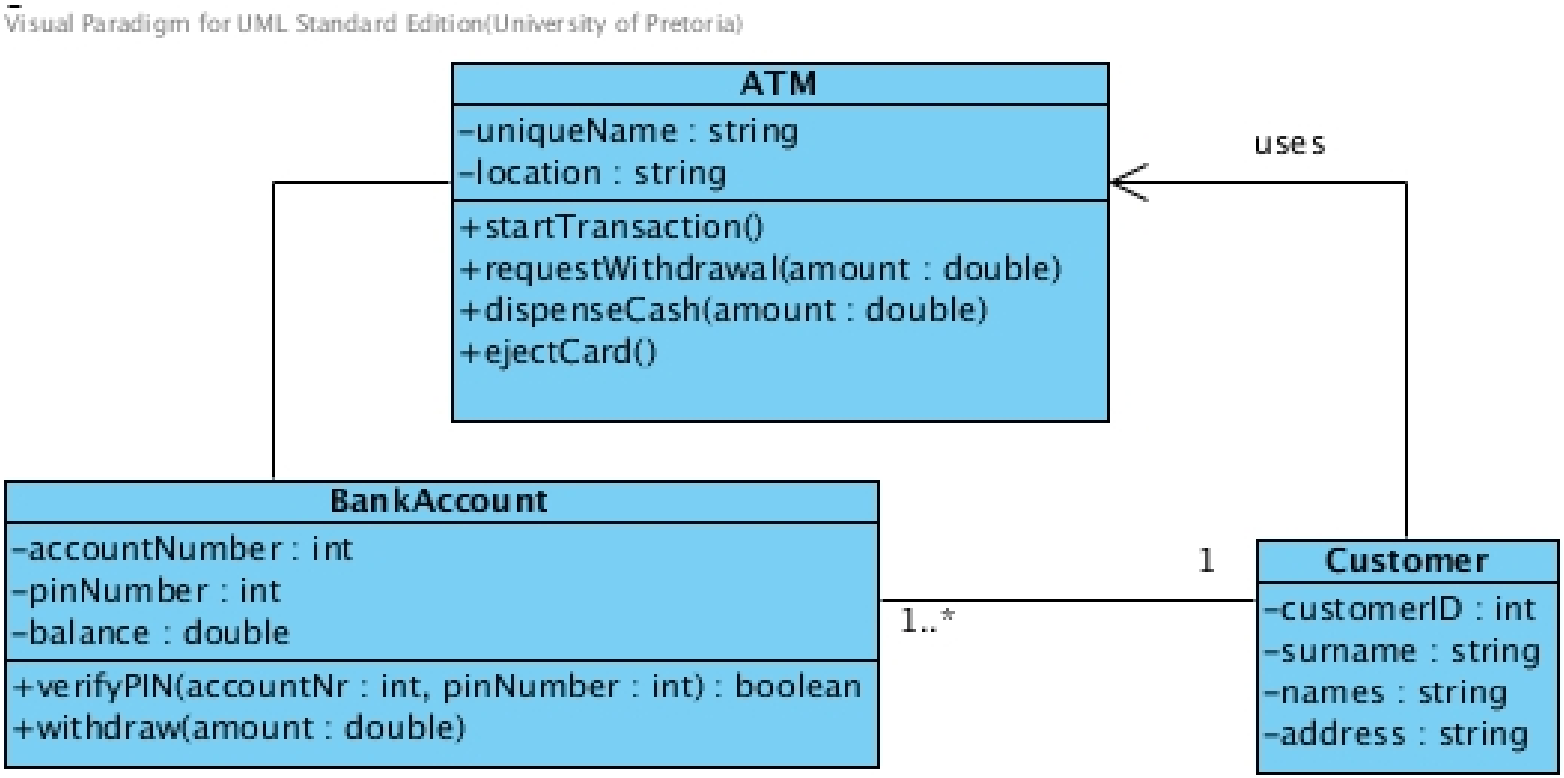
Composition

Navigatibility

Multiplicity

Multiplicity	**Meaning**
$n..n$	Exactly n
$0..n$	0 to n
$0..*$	0 or more
$m..n$	At least m and at most n

Multiplicity cont.


Visual Paradigm for UML Standard Edition(University of Pretoria)
ATM
-uniqueName : string
-location : string
+startTransaction()
+requestWithdrawal(amount : double)
+dispenseCash(amount : double)
+ejectCard()
uses
BankAccount
-accountNumber : int
-pinNumber : int
-balance : double
+verifyPIN(accountNr : int, pinNumber : int) : boolean
+withdraw(amount : double)
1
1..*
Customer
-customerID : int
-surname : string
-names : string
-address : string

Interface SiteSearch is used (required) by SearchController

http://www.uml-diagrams.org