

S. E

ASSIGNMENT

Submitted by

VINCE JOSEPH

Roll NO: 59

S3 MEA

1.)

Use case Diagrams

The most important aspect to model a system is to capture its dynamic behavior.

Dynamic behavior implies the behavior of sm when it is running.

In UML, there are 5 diagrams available to model the dynamic nature and use case diagram is one of them. For a dynamic behavior analysis, we need some internal or external factors for making interaction.

So, a use case diagram consists of internal and external agents called actors, use cases and their relationships. A single use case diagram captures a particular functionality of a sm. So we need a no. of use case diagrams to illustrate a whole sm.

usecase diagram - purposes

In brief purposes can be listed as:

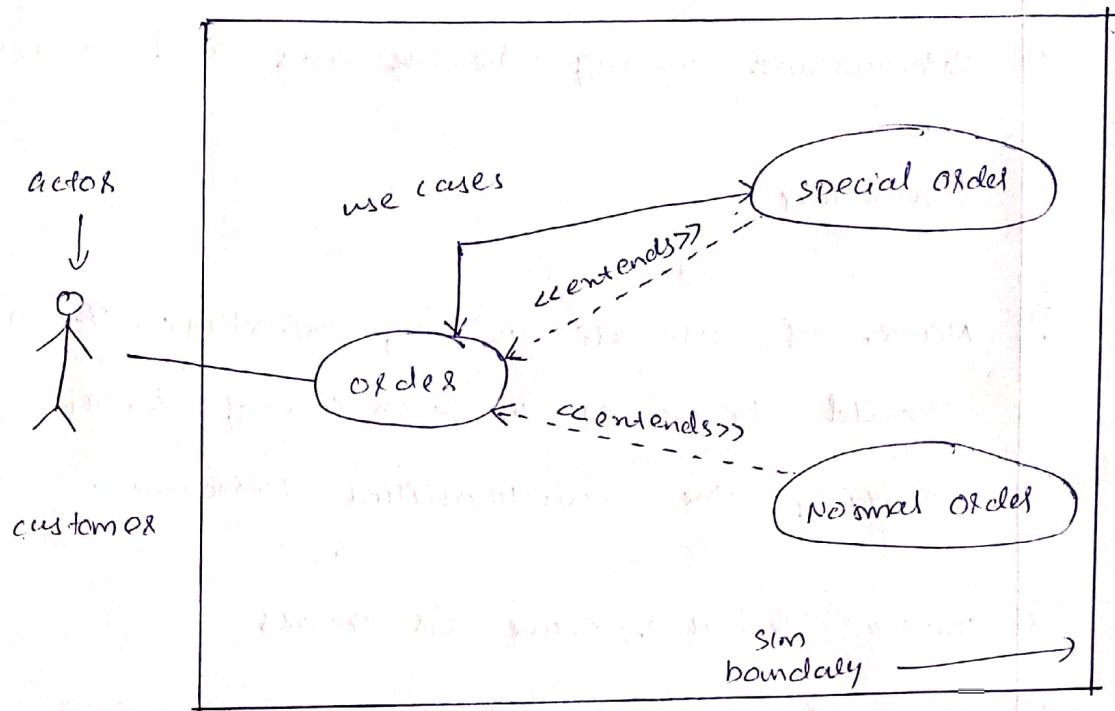
- i) to gather the requirements of a system,
- ii) to get an outside view of a sm.
- iii) to identify internal and external factors influencing the sm.
- iv) to show interaction among the requirements and actors.

Where to use this usecase diagrams

Usecase diagrams specify the events of a sm and their flows. But not describes how they are implemented; Somewhat like a black box where ^{only} input, output and function is known.

These diagrams are used at a very high level of design. This high level design is refined again and again to get a complete and ~~pr~~ practical picture of the sm. A well structured usecase also describes the precondition post condition and exceptions. These extra elements are used to make test cases when testing sm.

use case Diagram drawing



* use case diagram for order mgmt sim.

To draw a use case diagram, first analyze the requirements of the sim, then the functionalities are captured in use cases.

The second thing is to analyze the actors in a use case diagram. Actors can be a human or internal applications.

When we are planning to draw a use case diagram, we should have the following items identified:

- a) Functionalities to be represented.
- b) actors
- c) Relationships among the use cases and actors.

Guidelines

- a) Name of use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- b) Give a suitable name for actors
- c) show relationships and dependencies clearly
- d) Don't try to include all type of relationships.
- e) use notes whenever required to clarify something.

2.) Class Diagrams

1. This is a static diagram, represents static view of an application. It's used for visualizing, describing and documenting different aspects of a system and also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. These are the

Only diagrams which can be mapped directly with object oriented languages.

Purpose of class diagrams

- a) Analysis and design of the static view of an application.
- b) Describe responsibilities of a sm.
- c) Base for component and deployment diagrams
- d) Forward and reverse engineering.

How to draw

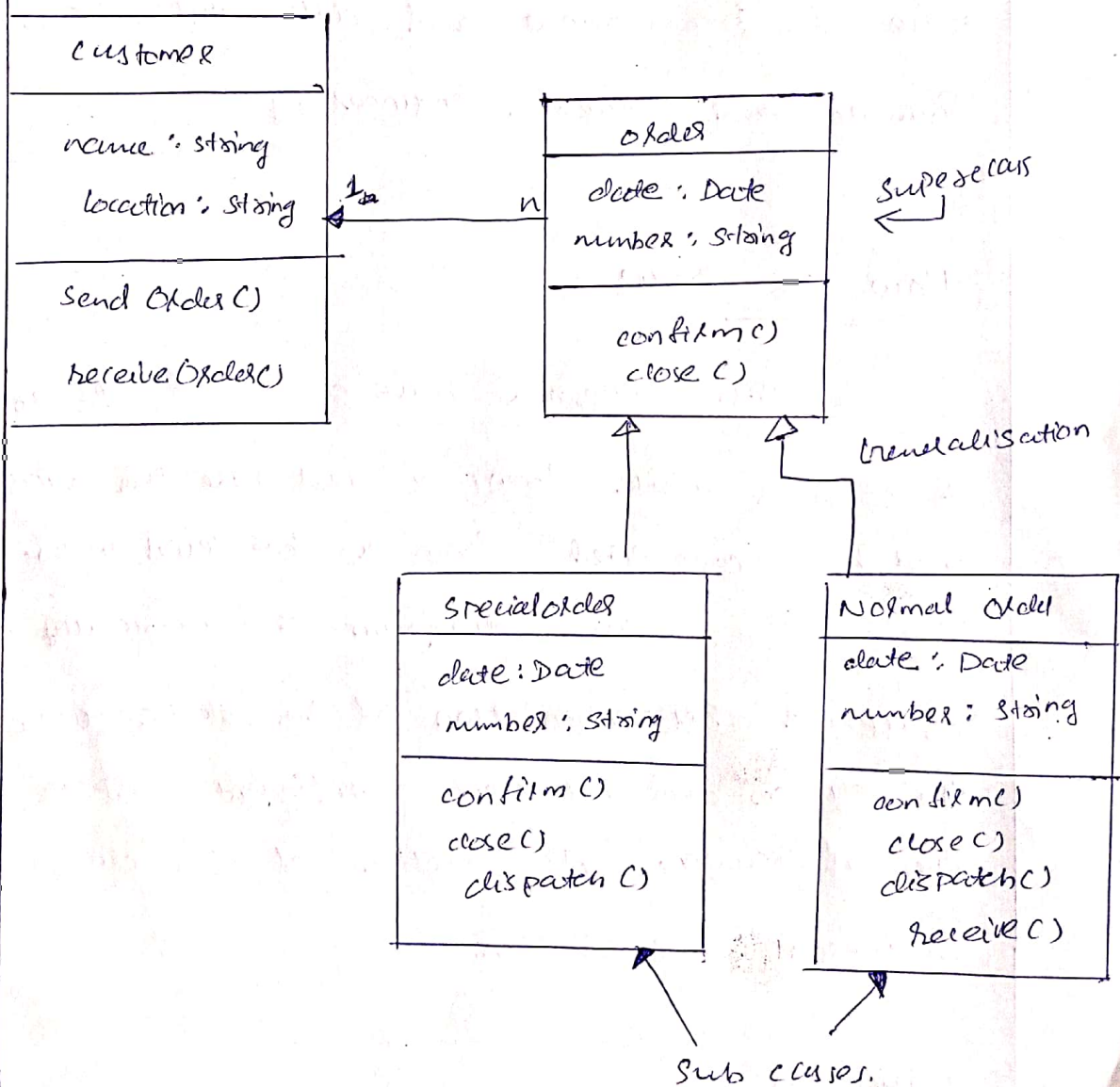
class diagrams have a lot of properties to consider while drawing but here the diagram will be considered from a top level view.

class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of class diagrams represent whole sm.

Guidelines

- a) Name of class diagram should be meaningful.

- *) Each element and their relationships should be identified in advance.
- *) Responsibility of each class should be specified clearly identified.
- *) use notes whenever required to describe some aspect of the diagram.



This is a class diagram for ordering an item.

(6)