Unit 2 – Design Patterns

Presentation by:
Ms. J.K. Josephine Julina
Assistant Professor
Department of Information Technology
SSN College of Engineering

Overview

DESIGN PATTERNS

GRASP: Designing objects with responsibilities – Creator – Information expert – Low Coupling – High Cohesion – Controller - Design Patterns – creational - factory method - structural – Bridge – Adapter -behavioral – Strategy – observer.

GRASP

GRASP stands for ***General Responsibility Assignment Software Patterns***

These are not ‘design patterns’, rather fundamental principles of object design

GRASP patterns focus on one of the most important aspects of object design, assigning responsibilities to classes.

GRASP patterns do not address architectural design

Characteristics of Good patterns

- It solves a problem
- It is a proven concept
- The solution isn't obvious
- It describes a relationship
- The pattern has a significant human component

Types of patterns

Architectural Patterns

Expresses a fundamental structural organization or schema for software systems.

Design Patterns

Provides a scheme for refining the subsystems or components of a software system, or the relationships between them.

Idioms

An idiom describes how to implement particular aspects of components or the relationships between them using the features of the given language.

- Information Expert
- Creator
- Low Coupling
- High Cohesion
- Controller
- Indirection
- Pure Fabrication
- Polymorphism
- Protected Variations

Object design

A simple definition (too simple☺):

- In the analysis part of the current and previous iterations you have
  - Identified use cases and created use case descriptions to get the requirements
  - Created and refined the domain concept model
- Now in order to make a piece of object design you
  - Assign methods to software classes
  - Design how the classes collaborate (i.e. send messages) in order to fulfill the functionality stated in the use cases.

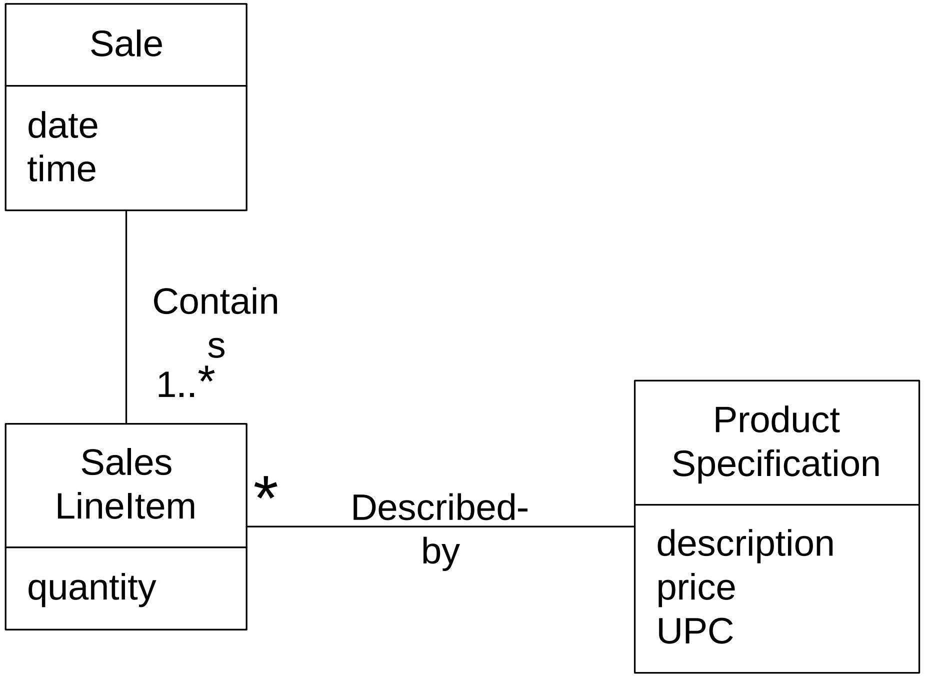
Critical step

- Central tasks in design are:
  - Deciding what methods belong where
  - How the objects should interact

- *A use-case realization* describes how a particular use case is realized within the design model in terms of collaborating objects.
- Use-case realization work is a design activity, the design grows with every new use case realization.
- Interaction diagrams and patterns apply while doing use-case realizations

Example

Who is responsible for knowing the grand total of a sale in a typical Point of Sale application?



References:

- Applying UML and patterns
  Craig Larman

- Patterns in Java, volume 2
  Mark Grand