



UNIT I

UML DIAGRAMS

**GALGOTIAS
UNIVERSITY**

UML DIAGRAMS

OBJECTS

- An object is a real world entity in an object oriented environment that may have a physical or conceptual existence.
- Each object has
 - Identity that distinguishes it from other objects in the system.
 - It has a determined characteristic properties
 - Well defined behavior that represent externally visible activities performed by an object.
- Objects can be modeled according to the needs of the application.
- An object may have a physical existence, like a customer, a car etc: or a conceptual existence like a project, a process etc.

UML DIAGRAMS

Object-Oriented Analysis

An investigation of the problem (rather than how a solution is defined). During OO analysis, there is an emphasis on finding and describing the objects (or concepts) in the problem domain. Usually performed by a systems analyst.

- Object oriented analysis is a process of identifying classes that plays an important role in achieving system goals and requirements.
- The primary tasks in Object oriented analysis are
 - Identifying objects
 - Organizing the object by creating object model diagram
 - Defining the object attributes
 - Defining the object behaviors
 - Describing objects interactions

For example, concepts in a Library Information System include *Book*, and *Library*. High level views found in the application domain. Often called domain objects; entities.

UML DIAGRAMS

Design

Object oriented design (OOD) involves implementation of the conceptual produced during Object oriented analysis. Usually performed by an architect or designer

- An architect looks at overall system structure (at a high level)
- For example, an architect on a distributed system might decide how the software components will be distributed. An architect also looks at modularity, and non-functional requirements (such as performance and scalability).

A designer looks at system structure (at lower levels)

A designer will look at the classes that make up a module, and how they will interact to perform some function of the system

Emphasizes a conceptual solution that fulfills the requirements. Need to define software objects and how they collaborate to meet the requirements. For example, in the Library Information System, a *Book* software object may have a *title* attribute and a *getChapter* method. What are the methods needed to process the attributes?

Designs are implemented in a programming language. In the example, we will have a *Book* class in Java. From Design to Implementation

UML DIAGRAMS

Design Patterns

Patterns are themes that recur in many types of software systems

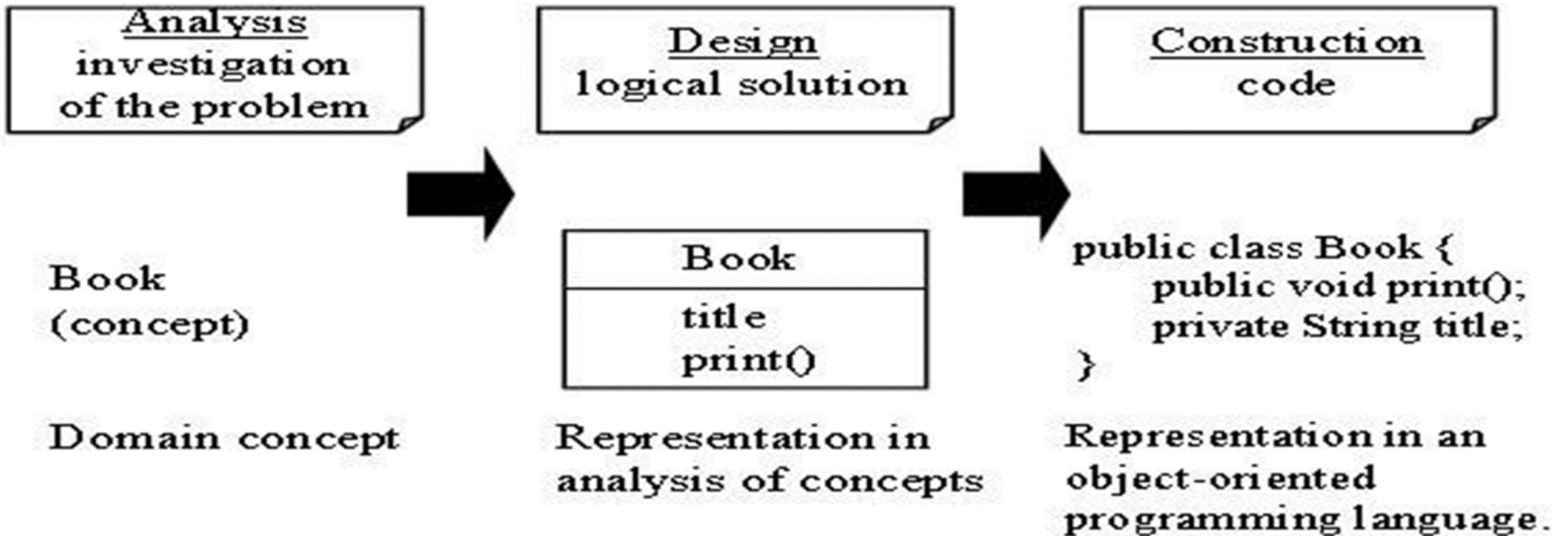
It is common for software systems (even those whose purpose is quite different) will share some common challenges. Often, the solution for the problem in one context can also be used in another context. Thus, the designs (and even implementations) can often be reused from other software systems.

Thinking in Objects and UML

The Unified Modeling Language (UML) is a standard diagramming notation; sometimes referred to as a blueprint. It is NOT OOA/OOD or a method. Only a notation for capturing objects and the relationships among objects (dependency; inheritance; realizes; aggregates,). UML is language independent. Analysis and design provide software blueprints captured in UML. Blueprints serve as a tool for thought and as a form of communication with others. But it is far more essential to think 'in terms of objects as providing services' and accommodating responsibilities.

UML DIAGRAMS

Thinking in Terms of Objects and UML



UML DIAGRAMS

Design (solution to requirements) assumes ‘ a robust requirements analysis has taken place. Use Cases are often used to capture stories of requirements and are often views as constituting‘ the functional requirements, but NOT the software quality factors (non-functional requirements). Use Cases are not specifically designed to be object- oriented, but rather are meant to capture how an application will be used. Many methods for capturing requirements. We will concentrate on Use Cases.

Our Approach:

We need a Requirements Analysis approach with OOA/OOD need to be practiced in a framework of a development process. We will adopt an agile approach (light weight, flexible) in the context of the Unified Process, which can be used as a sample iterative development process. Within this process, the principles can be discussed.

UML DIAGRAMS

Why the Unified Process:

The Unified Process is a popular iterative software development process. Iterative and evolutionary development involves relatively early programming and testing of a partial system, in repeated cycles. It typically also means that development starts before the exact software requirements have been specified in detail; Feedback (based on measurement) is used to clarify, correct and improve the evolving specification:

This is in complete contrast to what we usually mean by engineering!

What is the Unified Process?

The UP is very flexible and open and can include other practices from other methods such as Extreme Programming (XP) or Scrum for example. e.g. XP's test-driven development, refactoring can fit within a UP project; So can Scrum's daily meeting. Being pragmatic in adapting a particular process to your needs is an important skill: all projects are different

UML DIAGRAMS

The Rush to Code

Critical ability to develop is to think in terms of objects and to artfully assign responsibilities to software objects. Talk at great length about encapsulation and assigning methods to objects where the data is defined. One cannot design a solution if the requirements are not understood. One cannot implement the design if the design is faulty.

Analysis: - investigate the problem and the requirements.

- What is needed? Required functions? Investigate domain objects.
- Problem Domain
- The Whats of a system.
- Do the right thing (analysis)

UML DIAGRAMS

Design:

- Conceptual solution that meets requirements.
- Not an implementation
- E.g. Describe a database schema and software objects.
- Avoid the CRUD activities and commonly understood functionality.
- The Solution Domain
- The Hows of the system
- Do the thing right (design)
- OOA: we find and describe business objects or concepts in the problem domain
- OOD: we define how these software objects collaborate to meet the requirements.
 - Attributes and methods.
- OOP: Implementation: we implement the design objects in, say, Java, C++, C#, etc.
- Using the below model discuss the four activities

UML DIAGRAMS

Define use cases

Define domain model

Define interaction diagram

Define class diagram

GALGOTIAS
UNIVERSITY

UML DIAGRAMS

What is Analysis and Design

- Analysis:
 - investigate problem and its requirements; solution comes later
 - ask and answer questions
 - Example: Grade Book. Some questions?
 - *Requirements Analysis*
 - investigate requirements
 - *Object-oriented Analysis*
 - investigate objects used in and by domain

UML DIAGRAMS

A Short Example

Dicey: In which a program simulates a player tossing two dice. Define

Use Cases

- these are user scenarios, stories, goals
- Define a Domain Model

this is a description of the domain from the point of view of the objects involved

identify the concepts, attributes and associations

result is called the *domain model*

UML DIAGRAMS

REFERENCES:

1. Craig Larman, “Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development”, Third Edition, Pearson Education, 2005.
2. Simon Bennett, Steve Mc Robb and Ray Farmer, “Object Oriented Systems Analysis and Design Using UML”, Fourth Edition, Mc-Graw Hill Education, 2010.
3. Erich Gamma, and Richard Helm, Ralph Johnson, John Vlissides, “Design patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1995.
4. Martin Fowler, “UML Distilled: A Brief Guide to the Standard Object Modeling Language”, Third edition, Addison Wesley, 2003.
5. Paul C. Jorgensen, “Software Testing:- A Craftsman’s Approach”, Third Edition, Auerbach Publications, Taylor and Francis Group, 2008.

List of e-books

1. http://ebooks.lpude.in/computer_application/bca/term_6/DCAP308_OBJECT_ORIENTED_ANALYSIS_AND_DESIGN.pdf
2. http://home.vinhuni.edu.vn/cuongvcc/wpcontent/uploads/sites/109/2017/10/object_oriented_analysis_and_design_with_applications_2nd_edition_2788.pdf



Thank You