

System Analysis and Design

L01. Introduction

Topics

- Course Description
- Course Administration

Systems

课程中第一个关键字

- System is a very general term
 - Electrical, mechanical,...
- Computing Systems

我们关心

 - Computer systems doing computation (hardware and Software)
- Software System

课程涉及

 - A collection of software running on Common Computing System
- Software System can be viewed as a general System. So we are talking about **software systems** in our class.

课程名称中英文两个关键词

System Analysis and Design

- **Analysis** emphasizes an *investigation* of the problem and requirements, rather than a solution. 问题与需求的调研
"Analysis" is a broad term, best qualified, as in
 - *requirements analysis* (an investigation of the requirements) or
 - *object-oriented analysis* (an investigation of the domain objects). } 一般指
- **Design** emphasizes a *conceptual solution* (in software and hardware) that fulfills the requirements, rather than its **implementation**. 概念性的解决方法, 而非实现.

Methods for System Analysis and Design

- Structured (Functional) method
 - Object-oriented method
-
- We will talk about Object-oriented method

What is **Object-Oriented Analysis** and Design? 再一个关键词

- During **object-oriented analysis** there is an emphasis on *finding and describing the objects or concepts in the problem domain.* abstract
- During **object-oriented design** (or simply, object design) there is an emphasis on *defining software objects and how they collaborate to fulfill the requirements.*
- During **implementation or object-oriented programming**, design objects are implemented.

OOAD: Principles and Patterns

课程的一个主要内容

- There are many possible activities and artifacts in OOAD, and a wealth of principles and guidelines. One is “skillfully **assign responsibilities to software objects**”.
 - It is emphasized in this course as “Most Important Learning Goal”
 - This one thing influences the robustness, maintainability, and reusability of software components
- Certain tried-and-true solutions to design problems can be (and have been) expressed as best-practice principles, heuristics, or **patterns** named problem-solution formulas that codify exemplary design principles.
- This course, by teaching *how to apply patterns or principles*, supports quicker learning and skillful use of these fundamental object design idioms.

基本的对象设计风格

课程的另一个关键概念

Software Development Process

- Given many possible activities from requirements through to implementation, how should a developer or team proceed?
- Requirements analysis and OOAD needs to be presented and practiced in the context of some development process.
- A **software development process** describes an approach to building, deploying, and possibly maintaining software.

相同的另一概念:

Systems Development Life Cycle

- **The systems development life cycle (SDLC)**
 - SDLC as part of **software development process**
 - The SDLC has some fundamental phases (**planning, analysis, design, and implementation**).
 - Each phase is itself composed of a series of **steps**, which rely upon **techniques** that produce **deliverables**.
- Most of time, we mix software process and SDLC 我们几乎混用这两个概念 .

System Development Methodologies

与过程对应的
的一个概念

- System Development Methodologies

A methodology is a formalized approach to implementing the SDLC (i.e., it is a list of steps and deliverables.)

- Where do we start? --> Feasibility study (or, early requirements analysis)
- Define the problem --> Requirements analysis
- Design a solution --> Design

- This course is about methodologies for developing (software) systems!

Waterfall Development Process

一个具体的软件过程.

- In a **waterfall** (or sequential) lifecycle process there are attempts not that they will be. 不真实的期望.
- to define (in detail) all or most of the requirements before programming.
- to create a thorough design (or set of models) before programming.
- to define a "reliable" plan or schedule near the start

Sequential Process Phases in Waterfall Development Process

- **Communication**
 - project initiation;
 - requirements gathering
- **Planning**
 - estimating;
 - scheduling;
 - Tracking
- **Modeling**
 - analysis,
 - design
- **Construction**
 - code, test
- **Deployment**
 - delivery,
 - support,
 - feedback

理想化的顺序性的独立的阶段

Iterative and Evolutionary Development

课程的另一个重要的概念

- Development is in **short cycles, or iterations** 迭代
- Each one is **tested and integrated**
- Each one gives an **executable partial system**
- Feedback from each iteration leads to **refinement and adaptation** of the next. 导致下一个迭代 上的精化和修改
- An example of such a process is **the unified process (UP)**.

我们的课程主要讨论这种方法

Unified Modeling Language

在分析与设计使用的建模语言

- The Unified Modeling Language is a visual language for specifying, constructing and documenting the artifacts of systems
- Notation (the UML) is a simple, relatively trivial thing.
- Much more important: **Skill in designing with objects.**
 - Learning UML notation does not help
- The UML is *not*
 - a process or methodology
 - object-oriented analysis and design
 - guidelines for design

我们课程将使用 UML 进行建模。

Three Ways to Apply UML

三种方式(方法)

- **UML as sketch** Informal and incomplete diagrams created to explore difficult parts of the problem or solution space.
- **UML as blueprint** Relatively detailed design diagrams used either for
 - 1) reverse engineering, or
 - 2) code generation (forward engineering).
- **UML as programming language** Complete executable specification of a software system in UML.
- **Agile modeling** emphasizes *UML as sketch*; this is a common way to apply the UML, often with a high return on the investment of time (which is typically short).
- We will talk agile modeling in our course.

OO Modeling with UML

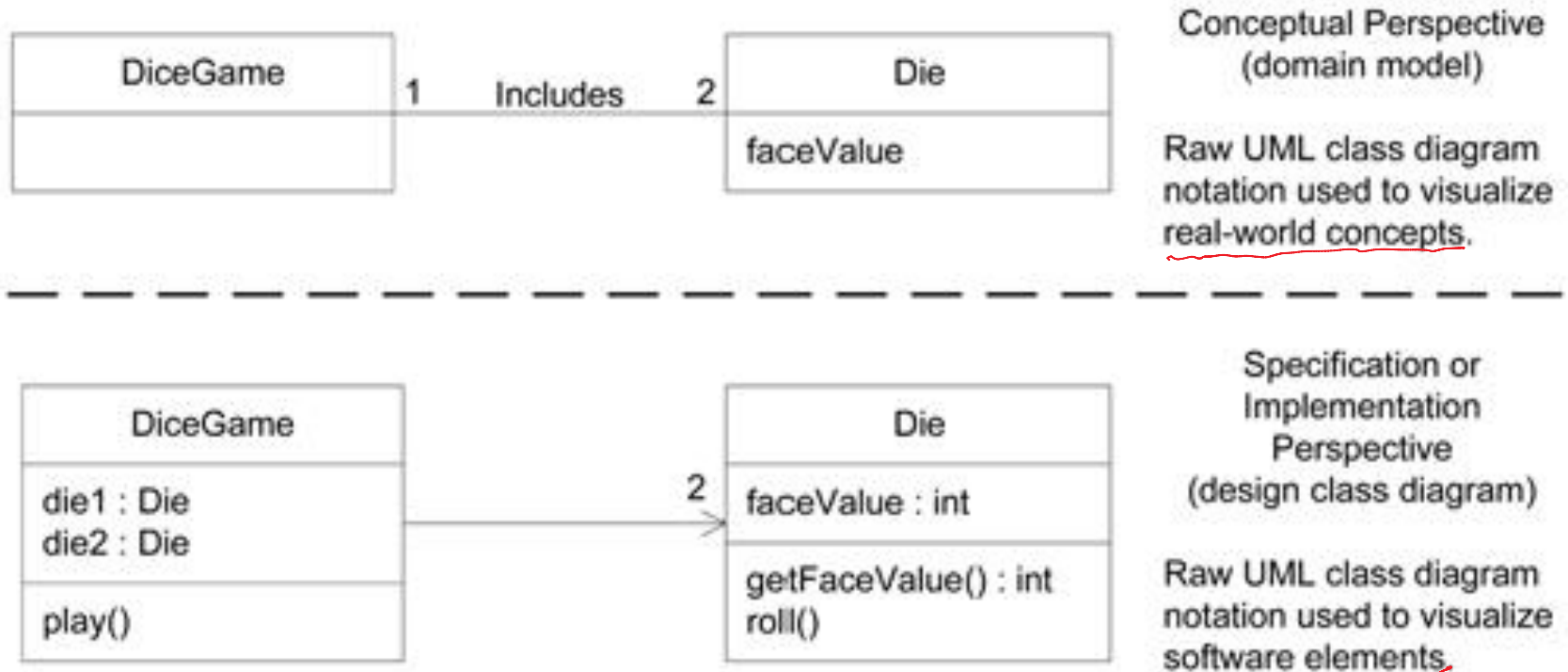
- Requirements and analysis:
 - Use case diagram
 - Interface Model
 - Business/Domain Object Model
 - Application Object Model
 - Object Interaction Model
 - Dynamic model
- Design
 - Design Object model
 - Design Object Interaction model
 - Design Dynamic model
- Implementation: Source code
- Testing: Test cases

三个层面

Three Perspectives of UML Models

1. **Conceptual perspective** the diagrams are 概念层 interpreted as describing things in a situation of the real world or domain of interest.
2. **Specification (software) perspective** the diagrams 规范描述层 describe software abstractions or components with specifications and interfaces, but no commitment to a particular implementation.
3. **Implementation (software) perspective** the 实现层 diagrams describe software implementations in a particular technology (such as Java).

Different perspectives with UML



The Meaning of "Class" in Different Perspectives

Class-related terms consistent with the UML and the UP

- **Conceptual class**—real-world concept or thing. A conceptual or essential perspective. The UP Domain Model contains conceptual classes. 概念层
- **Software class**—a class representing a specification or implementation perspective of a software component, regardless of the process or method. 软件层
- **Implementation class**—a class implemented in a specific OO language such as Java 实现层

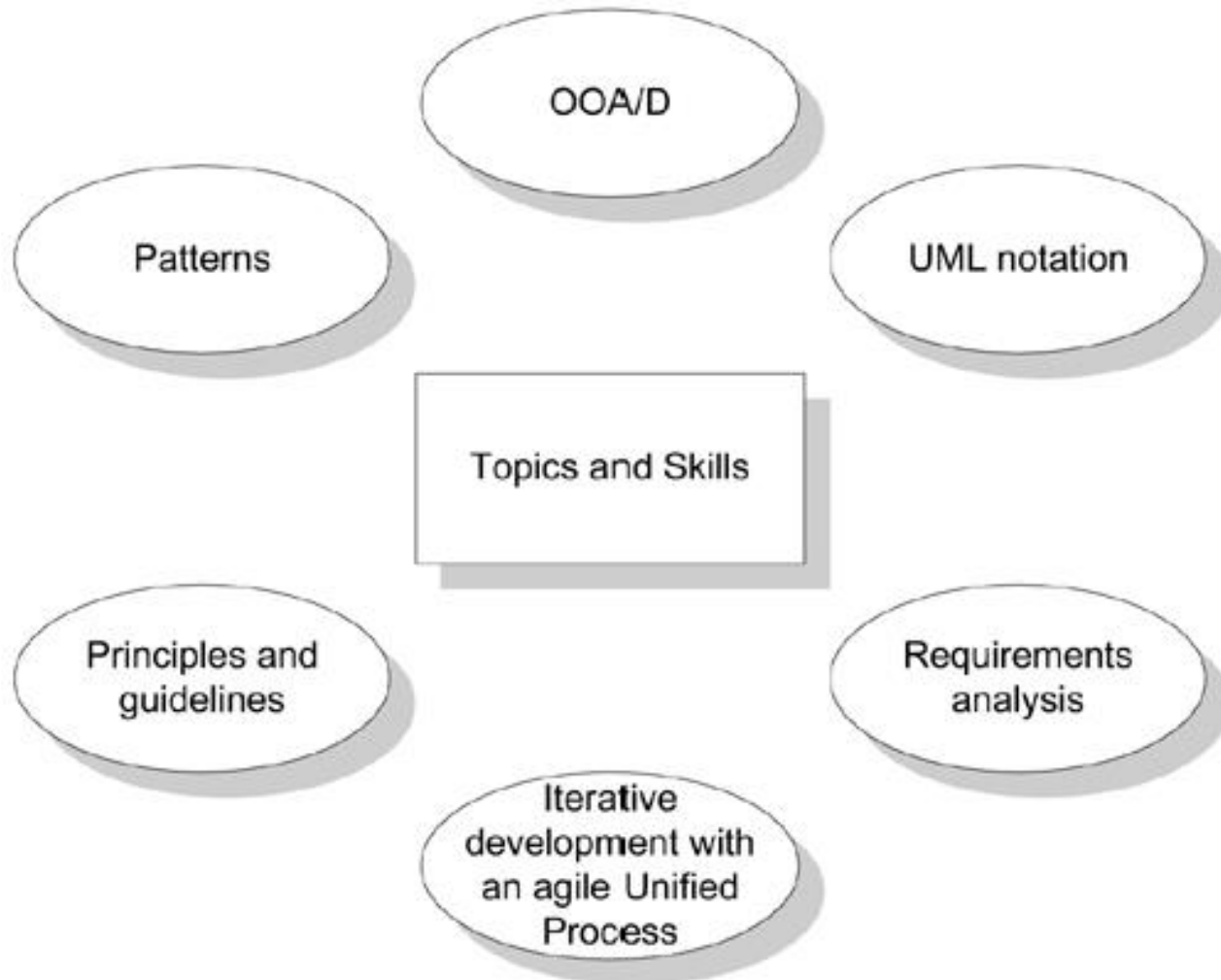
Course Subjects

- In the course, an **agile** (light, flexible) approach to the well-known **Unified Process** (UP) is used as the *sample* **iterative development process** within which the course topics are introduced.
- However, topics we talked in this course are independent of any particular process, and apply to many modern iterative, evolutionary, and agile methods.

What we will cover in the course

- Thinking in and design with objects
- OOD principles and Design patterns
- The agile methodology
- Apply UML (Visual Modeling)
- Analysis: An investigation of the requirements and the problem domain objects
 - evolutionary requirements analysis
 - writing use cases
- Design: Conceptual solution that fulfills the requirements

What we will cover in the course



This course helps a student

- Apply **principles and patterns** to create better object designs. 应用原理与模式
- Iteratively follow a set of common activities in analysis and design, based on an **agile approach** to the UP as an example. 敏捷分析与设计方法
- Create frequently used diagrams in the **UML notation**. 创建UML模型图

What You Will Do

课程中要求你亲自做分析与设计

- You will do analysis and design
 - You will write use cases
 - You will create a domain model
 - You will create class diagrams
 - You will write programs

Course Evaluation

- Homework (include midterm) (30%)
- Software analysis & design (Project) (30%)
- Final exam (40%)

Reference Books

Applying UML and Patterns:

An Introduction to Object-Oriented Analysis and Design and Iterative Development, Craig Larman, ISBN: 013 148 9062, Prentice-Hall, 2005.

Systems Analysis and Design

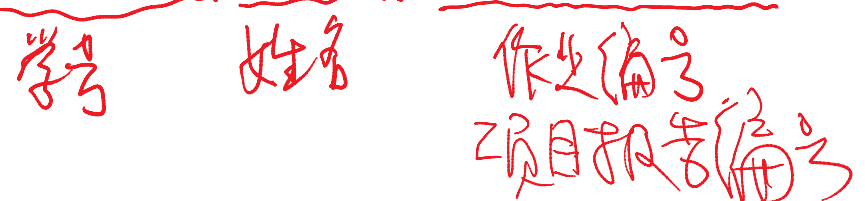
-- An Object-Oriented Approach with UML, 5th Edition, Alan Dennis, Barbara Haley Wixom, and David Tegarden, Wiley, 2014

Design Patterns Explained

A New Perspective on Object-Oriented Design, Second Edition
Alan Shalloway and James R. Trott, Addison Wesley, © 2005

- I will also be drawing on other resources throughout the semester

Submission

- Email: 616881900@qq.com
- Lecture notes
 - <https://pan.baidu.com/s/1vIBHbVBOaMKEnOMacUoLTA>
 - [密码:ubst](#)
- Submit with an attached file:
 - 111111.name.homework1.doc


Summary

- Systems
- System Analysis and Design
- Object-Oriented Analysis and Design
- System Development life Cycle (Software Process)
- System Development Methodologies
- Iterative and Evolutionary Development
- Modeling in System Development

希望可以复习今天的内容.