地理信息系统设计与开发

中国矿业大学-马骁

第一章 概述

软件工程：是一类求解软件的工程，应用计算机、数学、管理学等原理，借鉴传统工程的原则和方法，创建软件以达到提高质量、降低成本的目的。

是指导计算机软件开发和维护的一门学科。

IEEE定义：将系统化的、规范的、可度量的方法应用于软件的开发、运行和维护的过程，即将工程化应用于软件中。

软件：是计算机中与硬件相互依存的另一部分，包括程序、数据及其文档的完整集合。

GIS软件工程：是在整个GIS软件开发过程中，遵循一般软件开发的工程化原理和方法，并照顾到GIS软件开发的特殊规律和要求，对GIS软件建设的各个阶段进行工程化规范的一门技术。

软件工程的主要内容：

软件开发技术、软件开发管理

1. 软件开发方法

2. 软件开发过程
3. 软件工具和软件工程环境

> 软件生存周期：
>
> 定义时期：问题定义、可行性研究、需求分析
>
> 开发时期：总体设计、详细设计、编码和单元测试、综合测试
>
> 运行维护时期

软件生成周期划分

前期工程、设计工程、数据工程、工程实施、维护工程

软件生存周期模型规定了把生存周期划分成哪些阶段及各个阶段的执行顺序，也称过程模型。

瀑布模型、快速原型模型、增量模型

瀑布模型把开发过程划分为若干阶段，每个阶段相对独立，便于分工协作，每个阶段结束前，都从技术和管理两个角度进行审查，确认后开始下一阶段，是文档驱动的。

快速原型模型：首先建立一个能够反映用户需求的模型，让用户看下系统的全貌，哪些地方需要改进，然后对原型反复改进，建立符合用户要求的新系统。

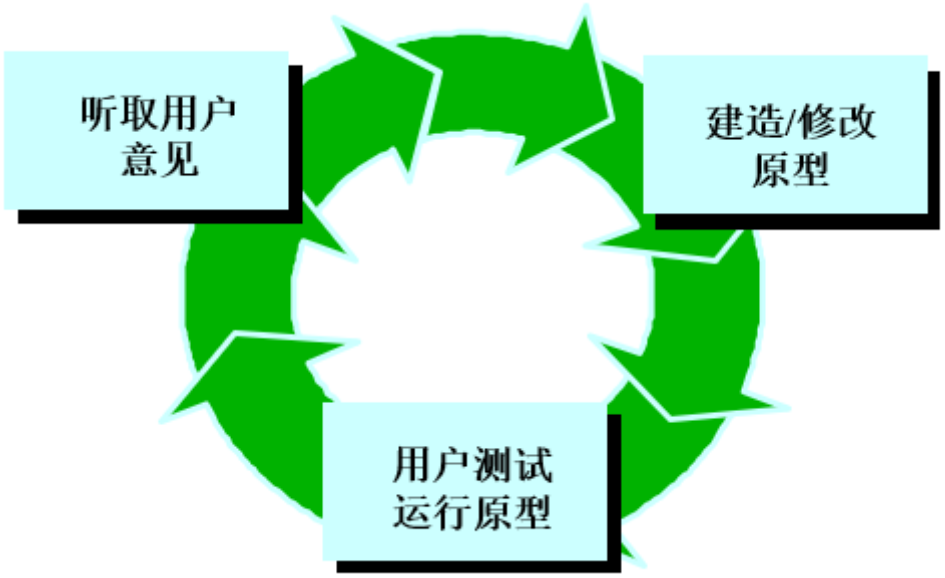


Figure 1-1

增量模型：融合了瀑布模型和原型实现的迭代特征，把软件产品作为一系列的增量构件来设计、编码、集成和测试，每个构建能够完成特定的功能。

敏捷过程：为了使软件开发团队具有高效工作和快速响应变化的能力，

- 个体交互胜过工具
- 可以工作的软件胜过面面俱到的文档
- 客户合作胜过合同谈判
- 响应变化胜过遵循计划

第二章 可行性研究

可行性研究的目的：用最小的代价在尽可能短的时间内确定该软件项目是否能够开发，是否值得开发。

由分析员完成，写出可行性研究报告，如果结论可行，则制定项目实施计划，不可行，则终止项目。

GIS可行性研究的影响因素：

- 技术因素
- 经济因素
- 社会因素
- 理论因素

> 可行性研究的任务：
>
> 首先进一步分析和澄清问题定义，并列举出对目标系统的约束和限制
>
> 分析员导出系统的逻辑模型
>
> 从逻辑模型出发，探索若干种解法，对每种解决方法都要研究它的可行性。

可行性研究的步骤：

- 确立项目规模和研究目标
- 研究正在运行的系统
- 建立新系统的高层逻辑模型
- 进一步定义问题
- 导出和评价各种方案
- 推荐可行的方案
- 编写可行性研究报告

项目组规模与项目组总生产率关系计算：

项目组规模与项目组总生产率的关系：

对于一个规模为P的项目组，项目组总生产率为：

$$L_{tot} = P(L - I(P-1)^r)$$

对于给定的一组L，I和r的值，总生产率L_{tot}是项目组规模P的函数。

L：不与任何人通信时的个人生产率；

I：生产率下降值；

r：是对通信路径数的度量；

Figure 2-1

第三章 需求分析

需求分析的任务：

- 建立分析模型
- 编写软件需求规格说明书

需求分析的步骤：

- 需求获取
- 需求提炼 （建立分析模型）
- 需求描述 （编写软件需求规格说明书）
- 需求验证

分析模型使描述软件需求的一组模型，包含信息流、处理功能、行为模型、及涉及约束等。

需求获取的内容：

功能需求、非功能需求

具体内容包括：

- 功能
- 性能
- 环境
- 界面
- 用户或人的因素
- 文档
- 数据
- 资源
- 安全保密
- 软件成本消耗与开发进度
- 质量保证

占主导化的两种分析模型：

- 结构化分析模型
- 面向对象分析模型

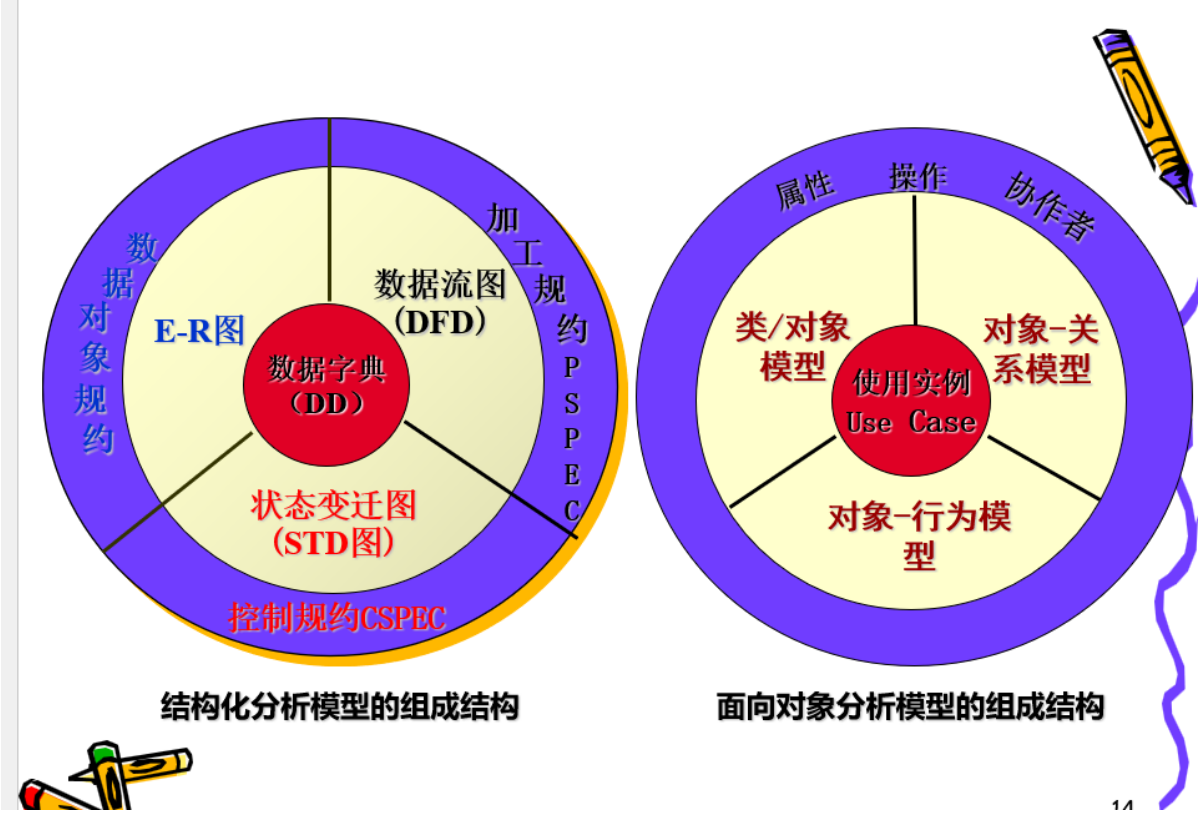


Figure 3-1

第四章 软件工程的分析方法

4.1 结构化分析方法

（1）数据流图的基本符号：

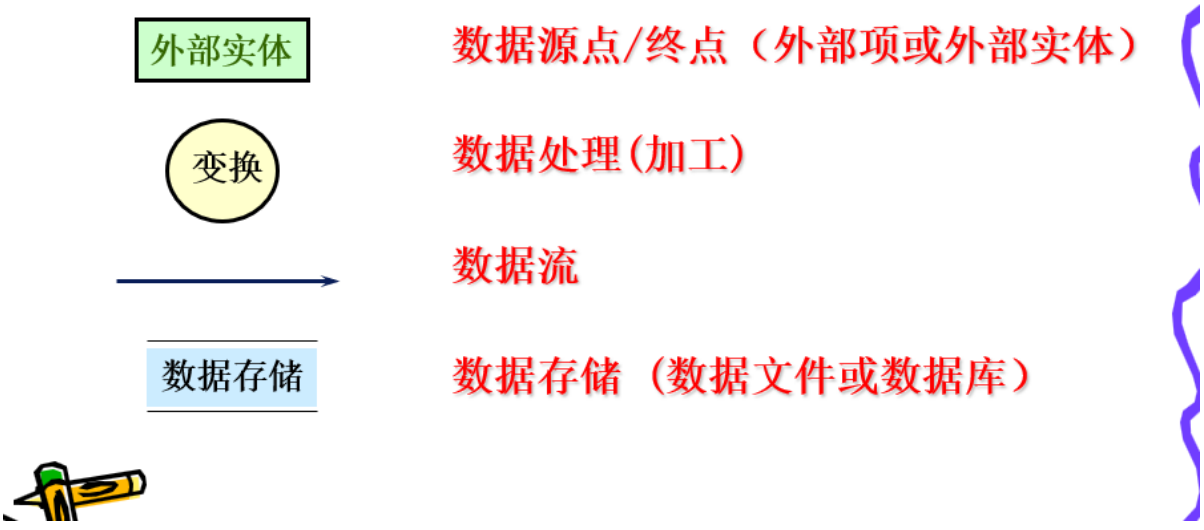


Figure 4-1

加工说明：应精确的描述一个加工该做什么，包括加工的激发条件、加工逻辑、优先级别、执行频率、出错处理等，最基本的是加工逻辑

结构化分析：结构化化分析是分析建模的一种方法，结构化分析方法建立模型的结果实际上是产生一组模型（数据、功能、行为），这一组模型是目标系统的逻辑表达，并且通过这些模型反应目标系统的完整需求。

分层数据流图：

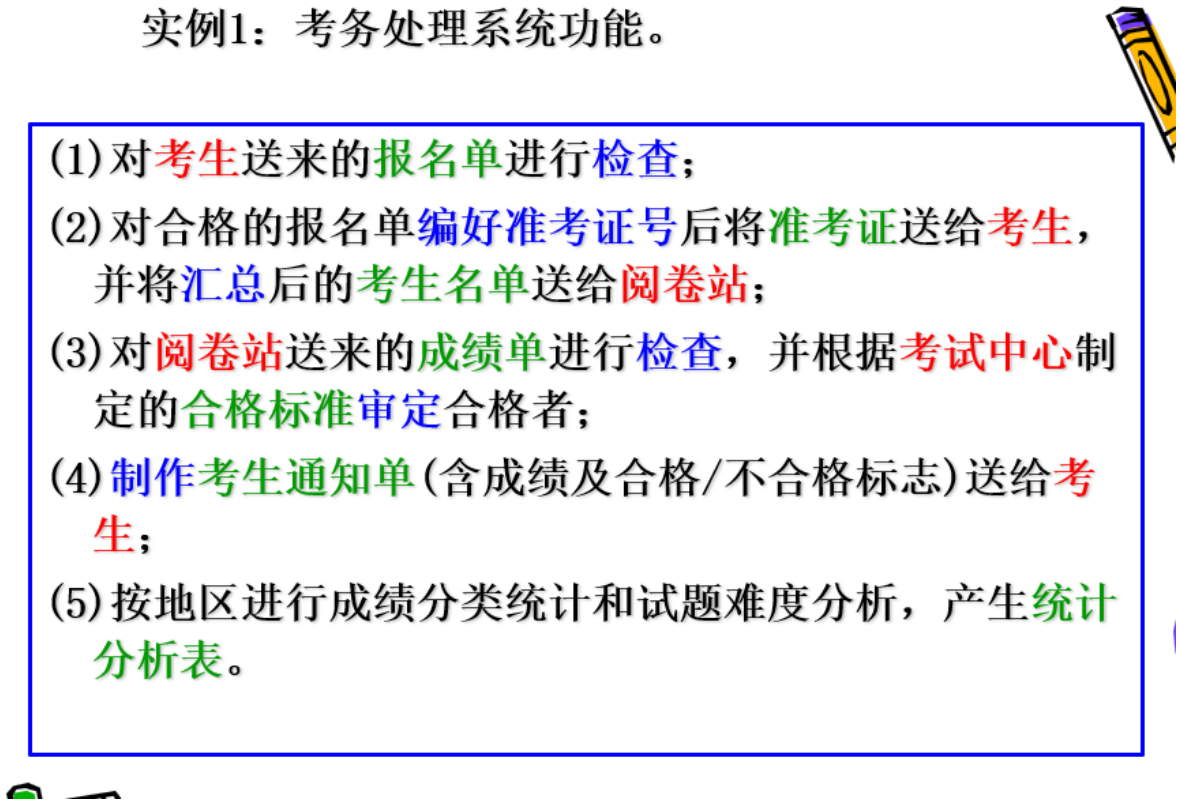


Figure 4-2

i>顶层数据流图

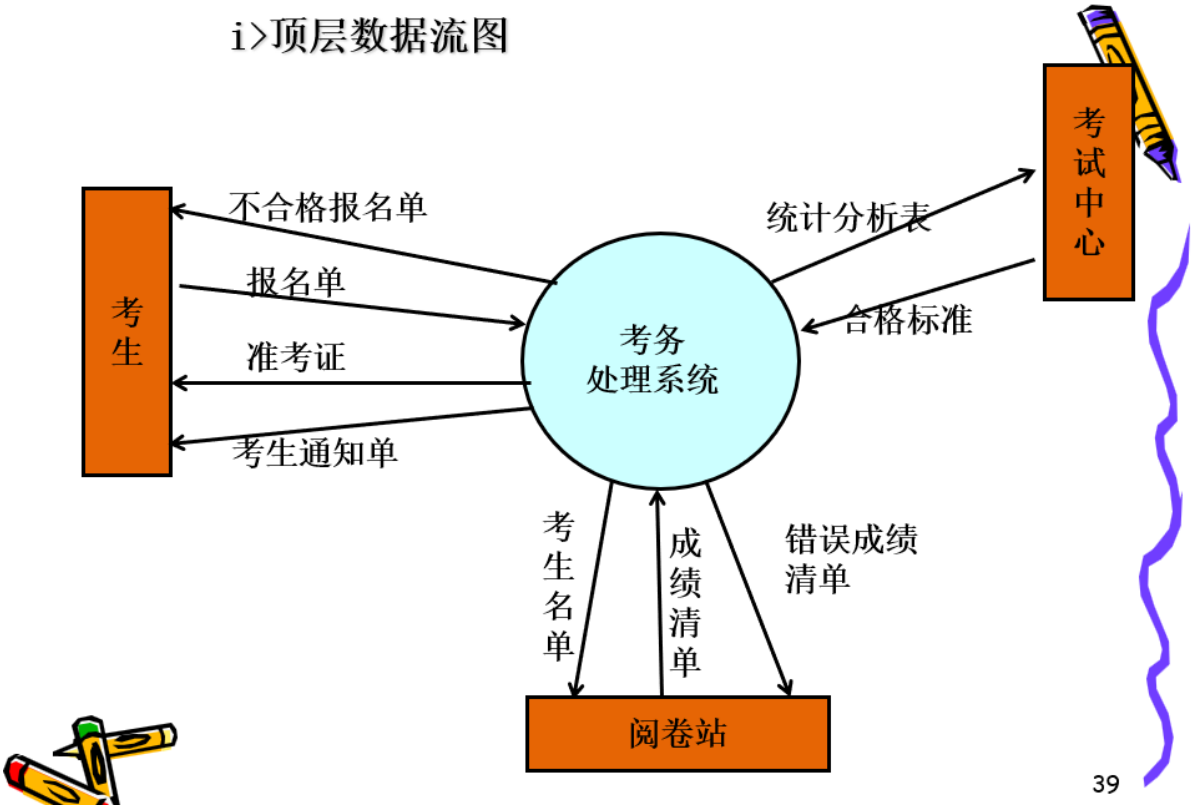


Figure 4-3

ii>一层数据流图

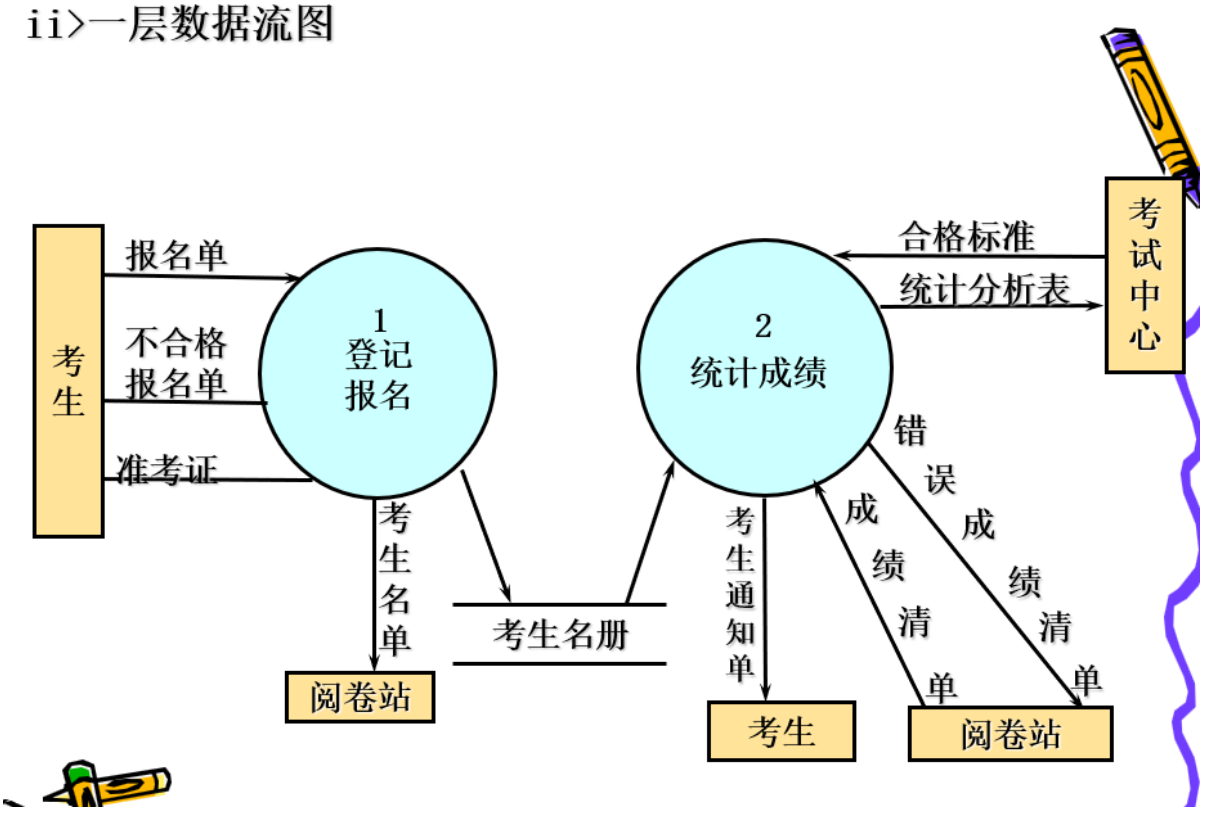


Figure 4-4

iii>二层数据流图

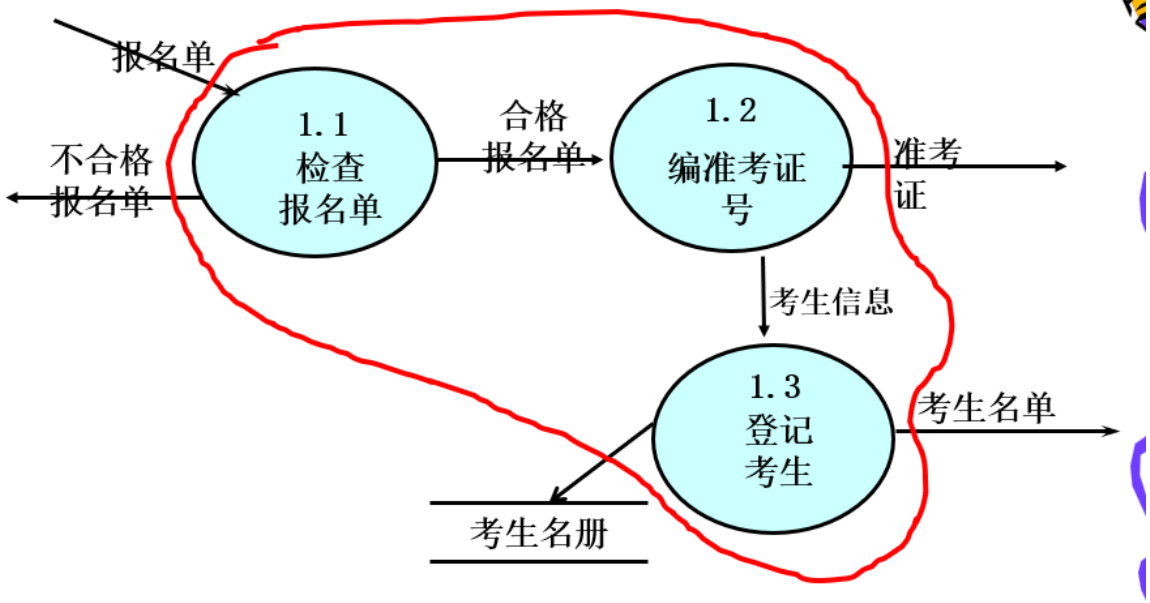


Figure 4-5

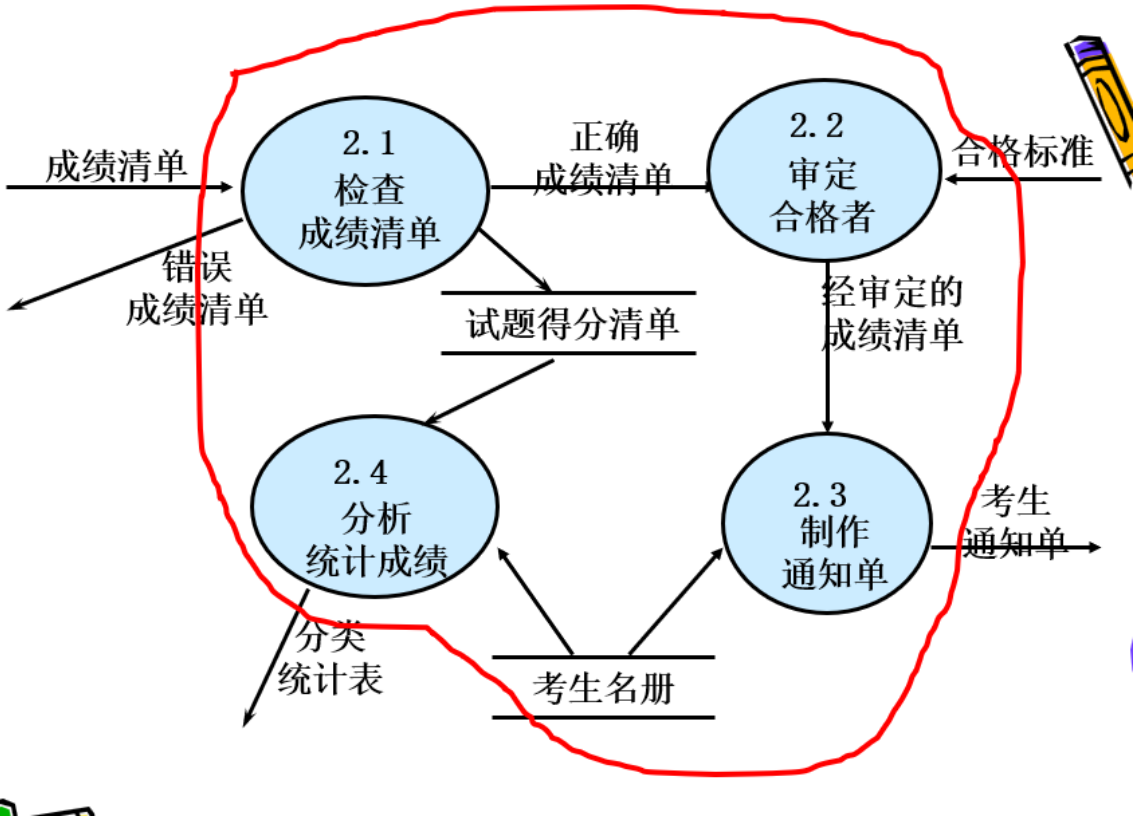


Figure 4-6

需求分析的复审技术：

4. 需求分析的复审

(1) 合理编号

- 子图号是父图中相应加工的编号；
- 子图中加工的编号由子图号、小数点、局部顺序号连接而成；

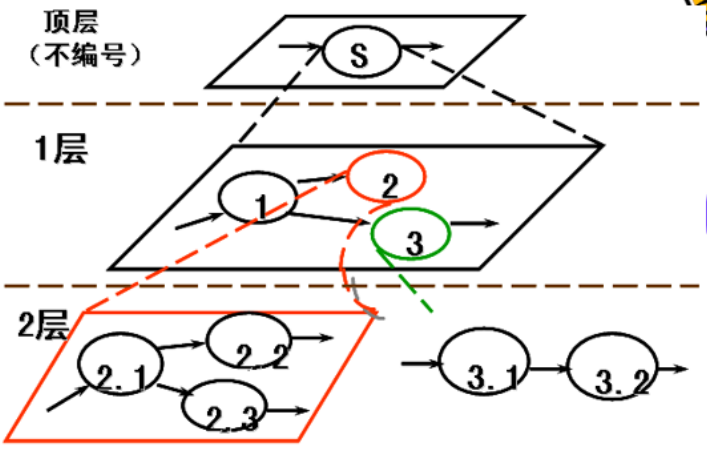
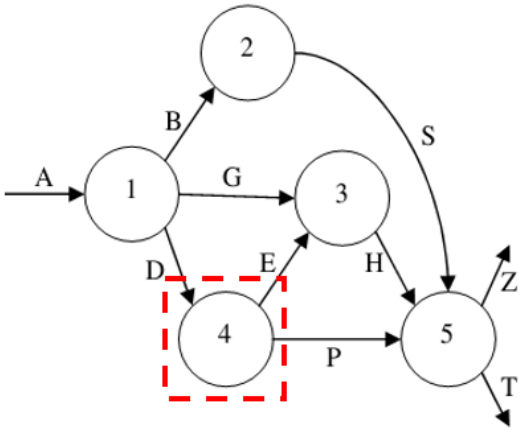
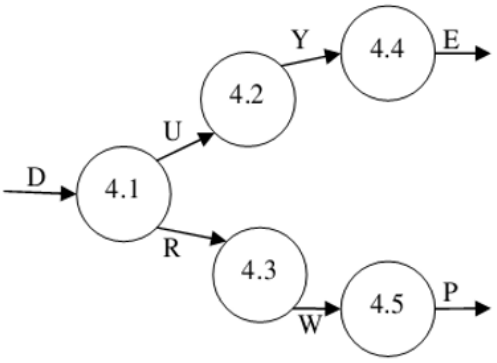


Figure 4-7

(2) 父图和子图的平衡

父图和子图的输入数据和输出数据应分别保持一致，称为父子平衡。





平衡的父图和子图

Figure 4-8

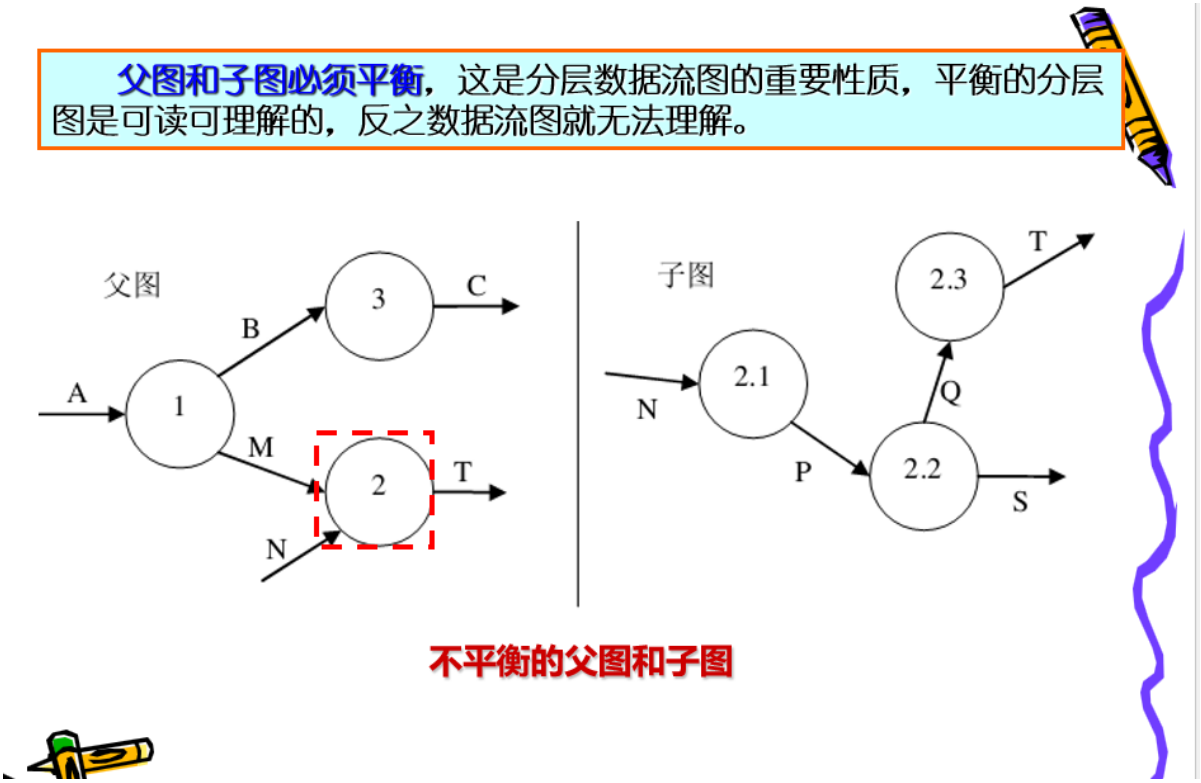


Figure 4-9

4.2 面向对象分析与建模

UML:统一建模语言（Unified Modeling Language）

UML是一种对软件密集型系统进行可视化、详述、构造、文档化的语言。

UML基本构造块：

- 事物（结构事物、行为事物、组织事物、注释事物）
- 关系（关联关系、依赖关系、泛化关系、实现关系）
- 图

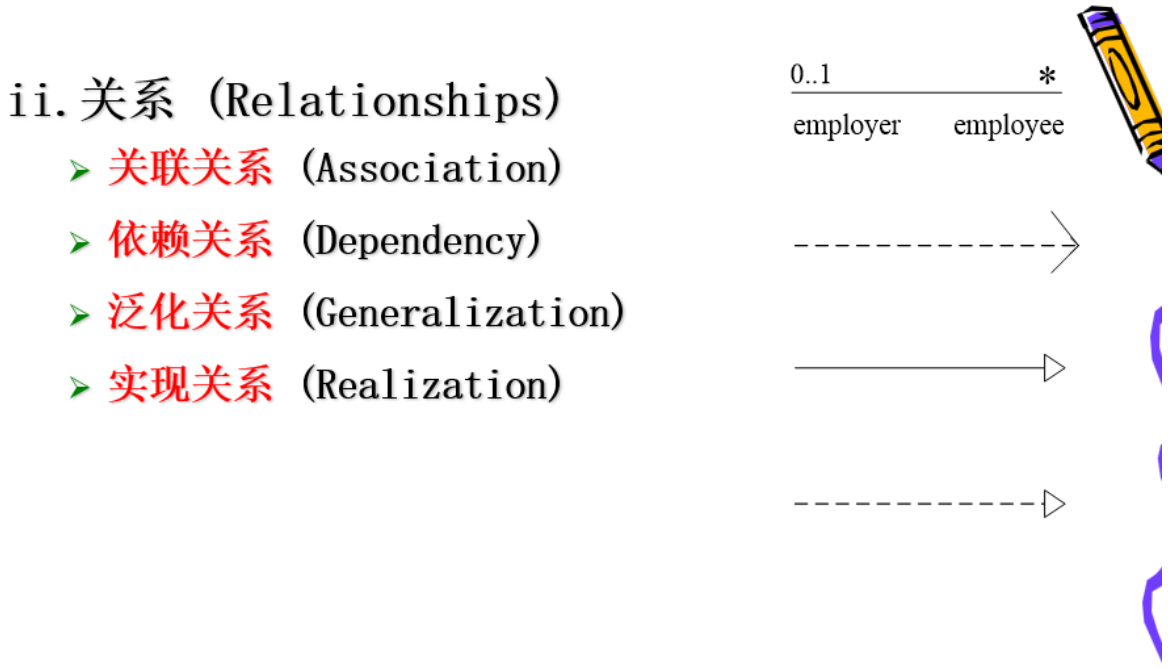


Figure 4-10

图：

- 用例图
- 类图
- 对象图
- 序列图
- 通信图
- 部署图
- 构件图
- 活动图 （程序流程图）
- 状态图

用例图是显示一组用例、参与者以及它们之间关系的图。

用例之间的关系有泛化、包含、扩展等。

某“授课系统”的功能为：学生能够浏览、查找、下载课件，还可以观看教学视频。

但是以上功能学生均需登录后才能正常使用。如果遗忘了密码，可使用“找回密码”功能。

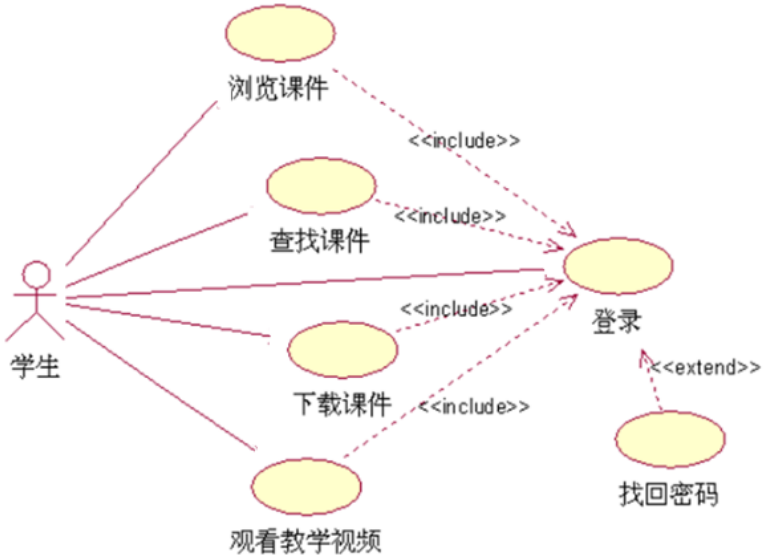
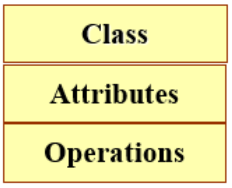


Figure 4-11

（1）UML中类的表示与命名

- UML中类的表示：



类

- UML中类的命名：

在UML中，类的命名分simple name和path name两种形式，其中simple name形式的类名就是简单的类的名字，而path name形式的类名还包括了包名。例如，下面是path name形式的类名：

Banking::CheckingAccount

Figure 4-12

主观题 10分 设置

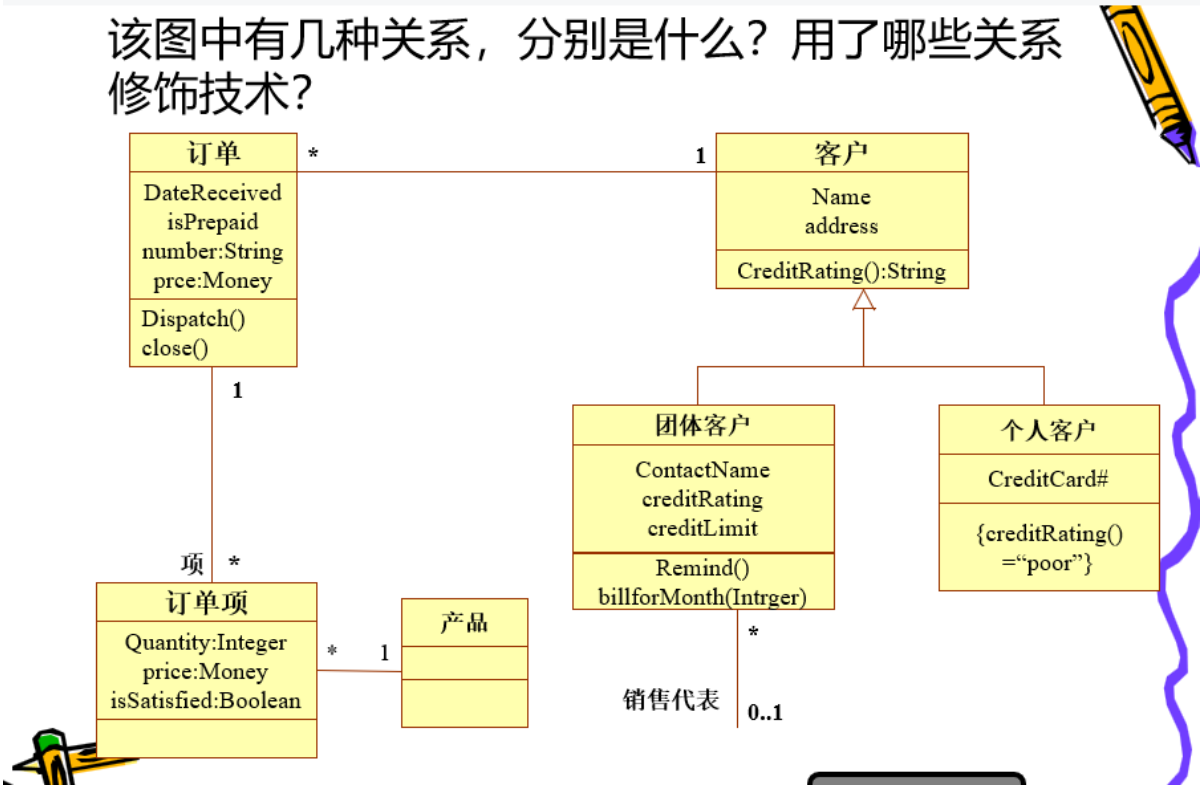


Figure 4-13

关联关系、依赖关系

导航管理，角色

序列图：考察单个用例内部若干对象的行为

（2）消息

消息的一般含义是触发接收消息的对象执行它的一个操作。

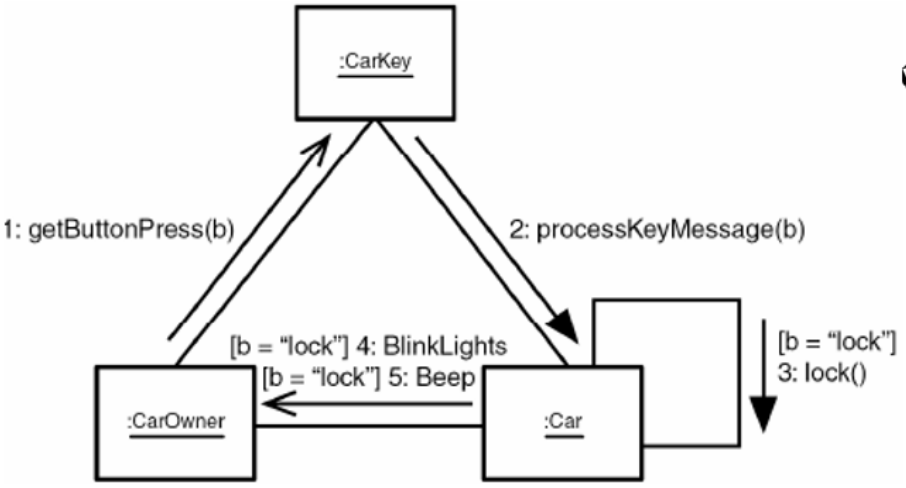
- 对象间的通信用对象的生命线之间的水平的消息线来表示；
- UML用从一条生命线开始到另一条生命线结束的箭头来表示一个消息。箭头的形状代表了消息的类型；
  - 同步消息（synchronous）：这是一个来自消息发送对象的请求，它被传递给消息的接受对象，它请求接收对象执行某种操作。通常，这需要发送者等待接收者来执行该操作； UML用一个带有实心箭头的实线来表示这种类型的消息。通常，这种情况包含了来自接收者的一个返回消息（是一条带有两条线的箭头的虚线），建模者经常忽略这个返回消息的符号；

  - 异步（asynchronous）消息：发送者把控制权转交给接收者，但并不等待操作完成；这种消息的符号是一个两条线的箭头；

50

Figure 4-14

示例：



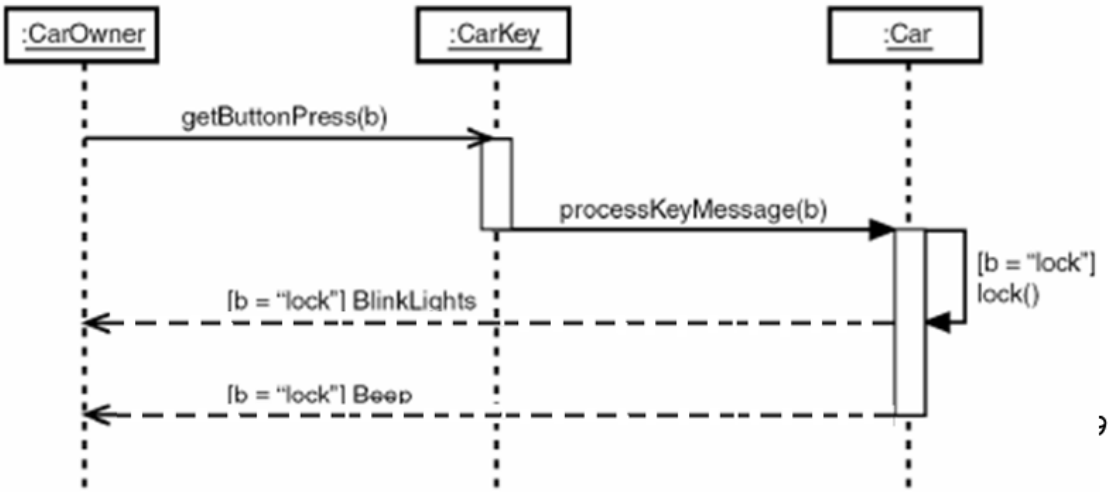


Figure 4-15

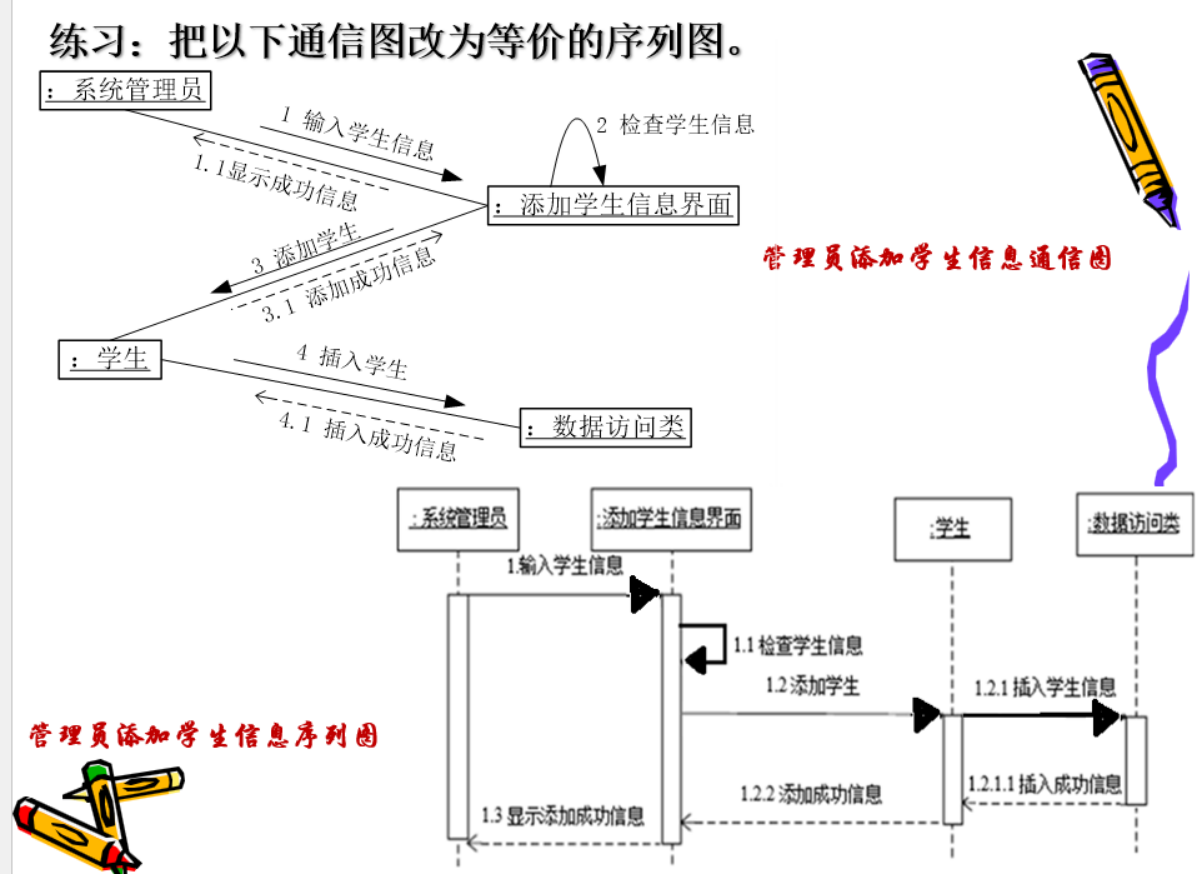


Figure 4-16

UML中的五种视图：

- 用户视图
- 结构视图
- 行为视图
- 实现视图
- 环境视图

结构化和面向对象方法的比较：

结构化方法数据与过程是分离的，软件系统是过程的集合

面向对象方法将对数据和处理方法封装为一个单元，系统是交互对象的集合

面向对象方法适合大型工程，强调对现实的模拟

结构化方法可维护性差

不同面向对象分析方法的相似步骤：

- 获取系统的用户需求
- 标识用例
- 使用基本需求作为指南，选择类和对象
- 为对象定义属性和操作
- 定义类的层次和结构
- 建造对象-关系模型
- 建造对象-行为模型
- 评审OOA模型

第五章 总体设计

系统设计是把一个系统需求变换成软件的表示过程。最初只是描绘系统的框架，然后进一步细化，在此框架中填入细节，把它加工成在程序上接近于源程序的软件表示。

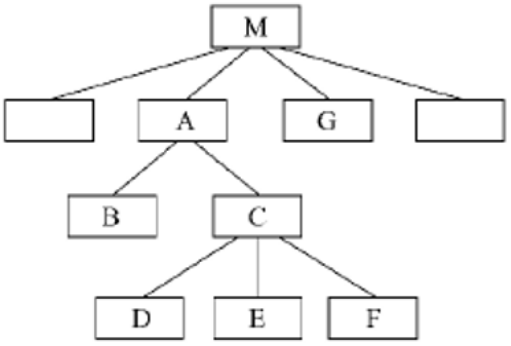
总体设计的任务：将系统需求转换为数据结构和软件体系结构。

模块指的是具有输入输出、逻辑功能、运行程序、内部数据四种属性的一组程序语句。

模块独立性的衡量标准：内聚、耦合

3. 模块的作用范围保持在该模块的控制范围内

- **作用域**是指该模块中一个判断所影响的所有其它模块；
- **控制域**指该模块本身以及所有直接或间接从属于它的模块；



在一个设计得很好的系统中，所有受判定影响的模块应该都从属于做出判定的那个模块，最好局限于做出判定的那个模块本身及它的直属下级模块。

Figure 5-1

总体设计工具：层次图，HIPO图

三层结构：表现层、业务逻辑层、数据访问层

3. B/S架构

- 三层体系结构突破了**客户/服务器两层模型**的限制，**将各种逻辑分别分布在三层结构中来实现**。这样可以将业务逻辑与表示逻辑和数据逻辑分开，从而减轻客户机和数据服务器的压力，达到有效平衡负载的目的，形成了一种新的计算模式——浏览器/服务器模式（B/S）。

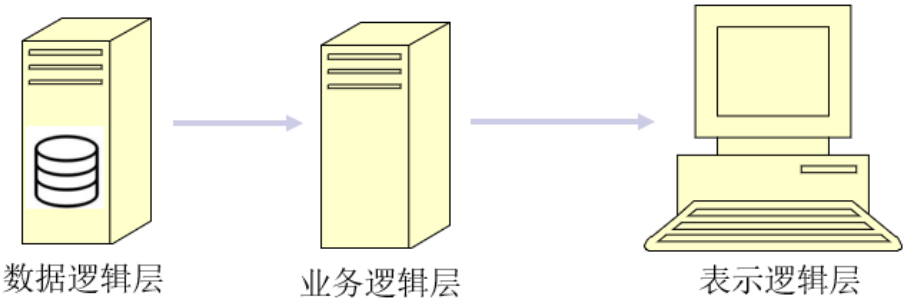


Figure 5-2

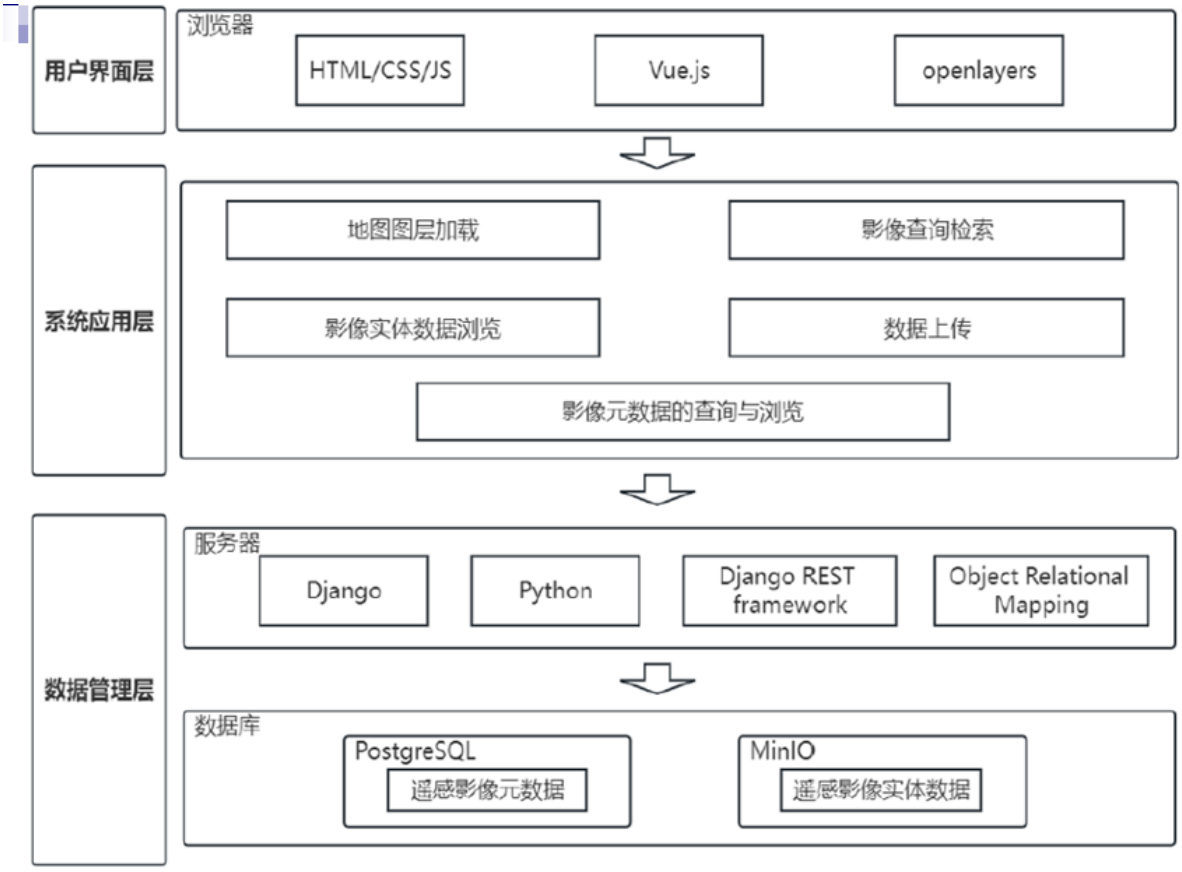


图 4-1 遥感影像存储检索系统架构图

Figure 5-3

地理空间数据库：是一种应用于地理信息处理和信息分析领域的数据库，主要存储对象是地理空间数据（空间数据，属性数据）

空间数据库管理方式

文档-关系数据库混合管理

全关系数据库管理系统

对象-关系数据库管理系统

面向对象数据库管理系统

1. 文件与关系数据库混合管理方案

- 属性数据建立在RDBMS上，数据存储和检索比较可靠、有效；几何数据采用图形文件管理，功能较弱，特别是在数据的安全性、一致性、完整性、并发控制方面，比商用数据库要逊色得多。
- 空间数据（属性数据与几何数据）分开存储，数据的完整性有可能遭到破坏。

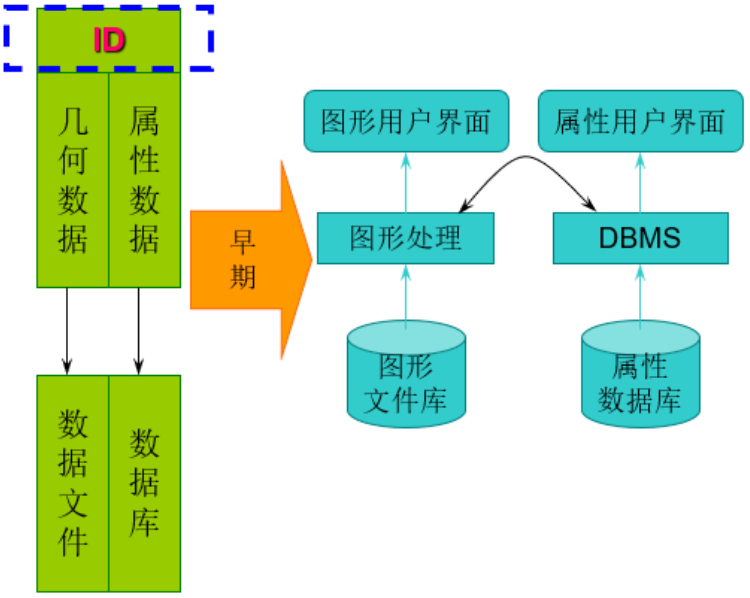


Figure 5-4

2. 全关系式数据库管理方案

- 属性数据、几何数据同时采用关系数据库进行管理；
- 空间数据和属性数据不必进行烦琐的连接，数据存取较快

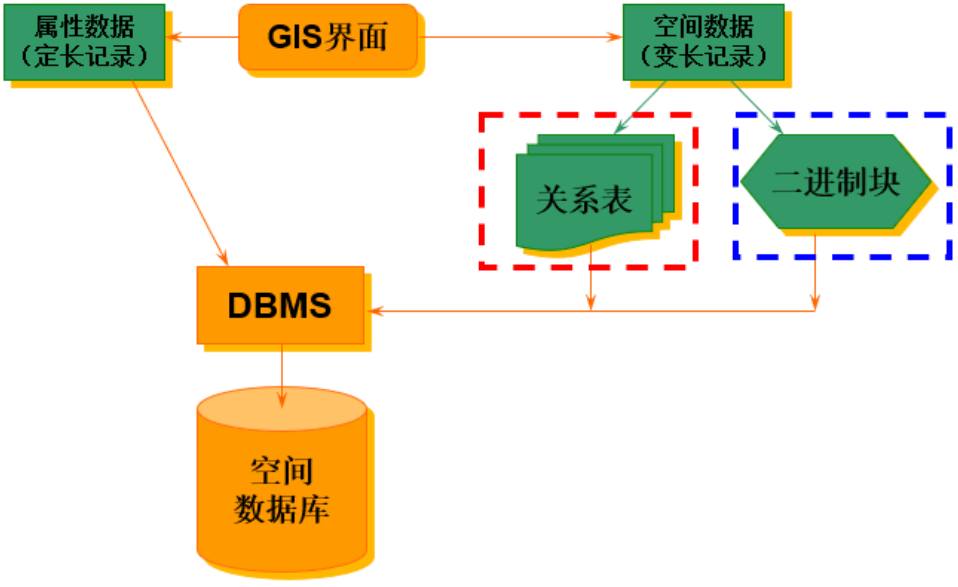


Figure 5-5

空间数据库引擎：

SDE：基于对特定的空间数据模型，在特定的数据库管理系统的基础上，提供对空间数据的存储、读取、检索等操作。以提供在次基础上的二次开发的程序功能集合。

- ArcSDE
- SuperMap SDX+
- Oracle Spatial
- PostGIS

第六章 系统详细设计

系统详细设计的主要任务：

- 为每个模块确定采用的算法
- 确定每一个模块使用的数据结构
- 确定模块接口的细节

详细设计的表达工具：

- 程序流程图
- 盒图
- 类程序设计语言

第七章 软件测试

软件测试：是为了发现错误而执行程序的过程

通常把一次程序执行需要的测试数据，称为一个“测试用例”，测试用例由测试输入数据和与之对应的预期输出结果两部分组成。

测试文档包括测试计划和测试报告。

测试报告主体是测试结果，包括测试项目名称，实测结果与期望结果比较，发现的问题，以及测试的效果。

第八章 软件维护

可维护性指维护人员为纠正软件系统出现的错误或缺陷，以及为满足新的要求而理解，修改，完善软件系统的难易程度。

二、影响软件可维护性的因素

- 可维护性指维护人员为纠正软件系统出现的错误或缺陷，以及为满足新的要求而理解、修改和完善软件系统的难易程度。
- 可维护性是支配软件开发的各个步骤的一个关键目标。
- 影响软件可维护性的因素主要有以下几个方面：
  - 可理解性；
  - 可修改性；
  - 可测试性；
  - 文档；
  - 软件的开发条件；
  - 软件的开发方法；

Figure 8-1