



# 《软件体系结构与设计》

## 作业一

班 级： 111171  
学 号： 20171000970  
学 生 姓 名： 董安宁  
指 导 教 师： 尚建嘎

中国地质大学地理与信息工程学院软件工程系

2019 年 9 月

# 第 1 章 软件体系结构基本概念

## 作 业

### 1. 简要说明为什么软件工程领域要引入软件体系结构?

随着软件系统规模和复杂性的成长，传统意义上数据结构+算法的程序已经无法设计和说明整个系统的结构。随着软件危机程度的加剧，软件体系结构应运而生，软件体系结构着眼于软件系统的全局组织形式，弥补了软件开发领域在工程上有余而在理论上不足的缺点，在较高层次上把握系统各部分之间的内在联系，将软件开发的焦点从成百上千的代码上转移到了颗粒较大的体系结构元素和其交互的设计上。有利于构件的重用和软件生产率的提高。

### 2. 给出一种你认为较恰当的软件体系结构定义?

构件：一组代码或是一个独立的程序

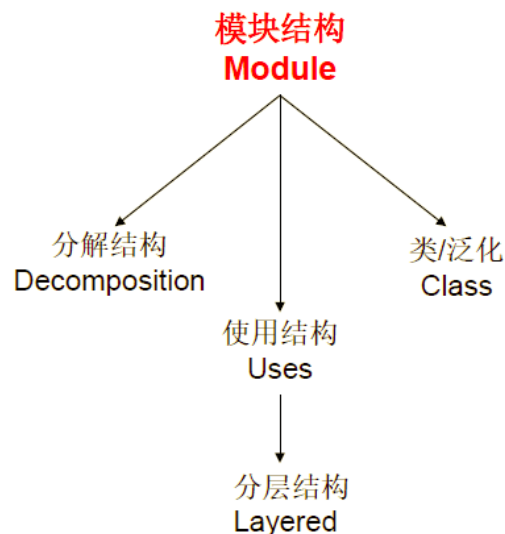
连接件：表示构件之间的相互作用，可以是过程调用、管道和消息等等

约束：为构件链接时候的条件

### 3. 软件体系结构中涉及哪几种结构？简述其含义，并举例说明。

模块结构 (Module)

系统如何被构造为一组代码或数据单元的决策



### 分解结构

这些单元是通过“子模块”关系将彼此关联起来的模块，展示了如何讲较大的模块递归地分解为较小的模块，直到他们足够小，很容易理解为止。

### 使用结构

单元之间通过“使用”关系这种特殊的一来关系实现彼此关联

### 分层结构

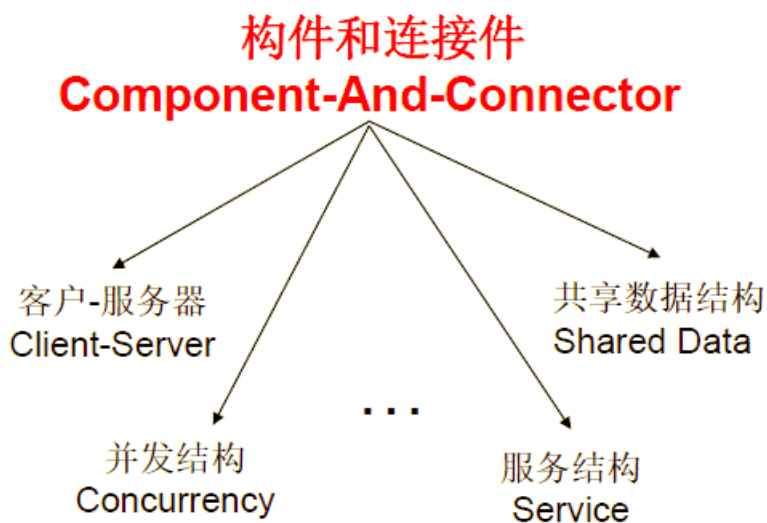
当以一种特定的方式小心滴控制该结构中的使用盥洗室，就出现了由层组成的系统，在该系统中，一个层就是相关功能的一个一致的集合。

### 类或泛化结构

将功能封装成类，以对象的形式调用。

## 构建和连接件结构（Component-And-Connector C&C）

系统如何被设计为一组具有运行时行为（构件）和交互（连接件）的元素



### 服务结构

系统结构单元是服务，服务之间通过一定的协作机制进行交互。

### 客户/服务器结构

构件是客户和服务器。连接件是协议和消息。

### 共享数据结构

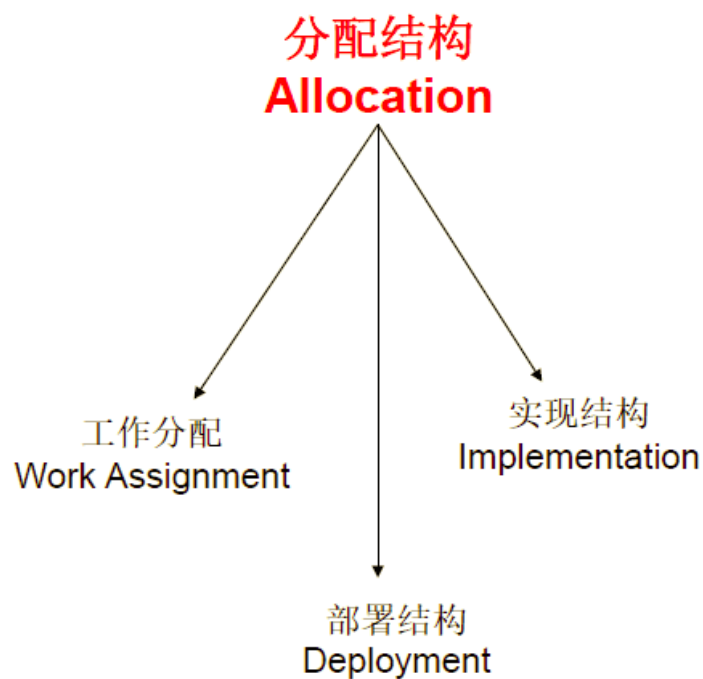
构件是作用在数据上的计算单元和数据存储单元，连接件则是提供的数  
据存取机制。

## 并发结构

结构单元是**组件**，连接件是他们之间的**通信机制**。组件被划分成逻辑线程。

## 分配结构（Allocation）

展示如何将来自于模块结构和 C&C 结构的单元映射到非软件结构（硬件、开发组和文件系统）



## 部署结构

软件是如何**分配到硬件**处理以及**通信元素**上的。

## 实现结构

该结构展示了**软件元素**（通常是模块）是如何**映射到**系统开发、继承或配置控制环境中的**文件结构上**（具体实现）

## 工作分配结构

该结构将实现和继承模块的**责任分配给**适当的**开发小组**

#### 4. 举例说明视图、视点、模型之间的区别和联系。

以盖房子为例，视点就是以某个房间为侧重点——比如健身房，保健室——去观察图纸中该房间是如何设计的，而忽略掉别的房间的图纸。视图就是我们所看到的那一部分图纸了。而模型就是通过一个统一的方法从不同视点观察各个房间，最终形成对整个房子的设计的一个完整的表达。

##### 视图

指的是视觉范围内看到的某物的一幅图景。**架构视图**是从某一视角或某一点上看到的系统所做的简化描述，描述中**涵盖**了系统的**某一特定方面**，**省略了与此方面无关的元素**

##### 视点

视点是一个**有关单个视图的规格说明**，用来描述系统中关注点划分问题。视图是基于某一视点对整个系统的一种表达。一个视图可由一个或多个架构模型组成。

##### 视图模型

一个视图是关于整个系统某一方面的表达，一个视图模型是指**一组用来构建系统或软件架构的相关视图的集合**，这样一组从不同视角表达系统的**视图组合在一起构成对系统比较完整的表达**。

##### 区别和联系

#### 5. 列举一些具有教材中所给出的软件体系结构风格的实际系统名称并简要说明？

##### 管道-过滤器风格(数据流)：

电子邮件应答系统：电子邮件的处理会经过几个阶段，每一个阶段对应着管道过滤器风格中的一个过滤器，邮件则作为传输数据在几个过滤器之间进行传递。

##### 过程调用风格：

几乎所有的程序或多或少地包括主程序/子程序调用风格，同时，在面向对象开发中，所有的程序也是自然地使用数据抽象和面向对象体系结构的风格。

### 独立结构风格：

调试器：独立结构风格的食指就是寄予时间的隐式调用风格，其主要特点是事件的触发者并不知道那些构建会被这些事件影响。编程环境中的调试器就是个很好的例子，当调试在断点处停下时，它只会声明事件，并不会关系哪些过程会启动，也不关心这些过程做什么操作。

### 层次结构：

Web 服务器：层次系统中，系统被组织成若干层次，每个层次由一系列构建组成，通过接口形成调用/返回的关系。如 Web 服务器分成表现层，业务层，数据层，即除了存储功能之外所有的功能和层都在一个服务器中，减少了跨服务器或服务器集群的物理边界进行层之间通信的性能开销。

## 6. 比较架构模式、设计模式、惯用法（idom）的区别和联系，举例说明。

以盖房子为例，架构模式就相当于搭建好了一幅骨架，剩下要做的就是往里面添砖加瓦。设计模式就是在添砖加瓦的过程中，每个施工队自己的组织安排。而惯用法就是每个施工队在添砖加瓦时候的具体细节，如谁谁干什么，怎么干。

### 架构模式

代表了模式系统中的最高抽象级别，有居于明确一个应用的基本结构。

### 设计模式

是中等规模的模式，他们在规模上比架构模式小，但又独立于特定编程语言或编程范式。

### 惯用法

具体针对一种编程语言的底层模式。惯用法描述如何使用给定语言的特征来实现组件的特殊方面或他们之间的关系

## 7. 常见的设计模式有那些，给出其中英文名称？

解释器模式（Interpreter）	迭代子模式（Iterator）	单例模式（Singleton）
建造者模式（Builder）	原型模式（Prototype）	命令模式（Command）
装饰模式（Decorator）	代理模式（Proxy）	外观模式（Facade）
桥接模式（Bridge）	组合模式（Composite）	享元模式（Flyweight）
策略模式（strategy）	观察者模式（Observer）	模板方法模式（Template Method）
访问者模式（Visitor）	备忘录模式（Memento）	状态模式（State）
中介者模式（Mediator）	工厂方法模式（Factory Method）	适配器模式（Adapter Pattern）
抽象工厂模式（Abstract Factory Pattern）	责任链模式（Chain of Responsibility）	