

# 软件体系结构学习报告

---

姓名：樊怡江 班级：19.2 学号：201920180260

## 1 软件体系结构的定义

---

软件体系结构没有一个公认的定义，简单来说，**软件体系结构就是充当需求 (Requirements) 和实现 (Implements)之间的桥梁**。由David Garlan & Dewane Perry 提出。

除此外，还需要掌握的是软件体系结构概念的提出者Mary Shaw & David Garlan所阐述的定义：**软件体系结构包括构成系统的设计元素的描述，设计元素的交互，设计元素组合的模式，以及在这些模式中的约束。**

综合其他的定义，可以得出如下公式：**软件体系结构 = 构件 (Component) + 连接件 (Connector) + 约束 (Constraint)**

- 构件

构件是可预制和可重用的软件部件，是组成体系结构的基本计算单元或数据存储单元。它可大可小，即可小到一个函数，又可大到一个子系统。构件应当满足SRP原则，即单一职责原则。

- 连接件

连接件也是可预制和可重用的软件部件，是构件之间的连接单元，比如共享变量的访问中共享变量就是一个连接件。而在现实世界的网络中，通信协议就充当了连接件。

- 约束

约束阐明了构件和连接件之间的关系。

## 2 软件体系结构的意义

---

### 2.1 软件体系结构在软件开发中的意义

- 软件体系结构是软件开发过程初期的产品，对于开发进度和软件质量的一切资金和劳务投入，可以获得最好的回报。
- 体系结构设计是形成的投资高汇报的重要因素。
- 正确有效的体系结构设计会给软件开发带来极大的便利。

### 2.2 软件体系结构的目标

- 外向目标：建立满足终端用户要求的系统需求。
- 内向目标：建立满足后期设计者需要以及易于系统实现、维护和扩展的系统部件构成

## 3 软件体系结构的研究范畴

---

- 软件体系结构描述语言和工具 (ADL)
- 产品线和标准：企业 -> 需求 -> 架构 -> 系统
- 软件体系结构风格以及风格的应用
- 体系结构文档化

## 4 软件体系结构的层次结构模型

---

- **计算机硬件（硬件基础层）**：软件运行的物质基础
- **软化的硬件层**：在硬件结构和性能抽象的基础上，实现硬件的操作和控制描述（处理器：状态和指令集合 中断：状态和中断服务）
- **基础控制描述层**：建立在高级程序语言描述上的纯粹软件描述层，包括了高级语言所支持的所有程序控制和数据描述概念（程序设计语言、结构化分析、面向对象分析设计）
- **资源和管理调度层**：在基础控制描述层建立的一切数据对象和操作，都需要在操作系统的协调和控制下才能实际的实现其设计的作用和功能（进程控制、分时系统、消息机制、I/O etc）
- **系统结构模式层**：最高层次的软件结构概念、属于体系结构风格或系统级别的设计模式、最高的抽象描述层（解释器、编译器、编辑器、管道/过滤器、黑板、C/S、B/S、框架 etc.）
- **应用层**：从纯粹应用领域出发所建立的系统结构概念（企业管理、公文处理、控制系统、CAD系统、ERP系统）

## 5 体系结构风格

---

### 5.1 主程序 - 子程序 风格

主程序-子程序风格是结构化程序设计的一种典型风格，从功能的观点设计系统，通过逐步分解和细化，形成整个系统的体系结构。

### 5.2 面向对象 风格

与前面的主程序 - 子程序风格比，面向对象风格中的构件变成了对象（不同于主程序、子程序的概念），并且连接的方式也发生了相应的变化。

**特点：**

- 系统被看作是对象的集合，每个对象都有一个它自己的功能集合；
- 数据及作用在数据上的操作被封装成抽象数据类型；
- 只通过接口与外界交互，内部的设计决策则被封装起来。

### 5.3 管道-过滤器风格

管道-过滤器风格把系统任务分成若干连续的处理步骤，这些步骤由通过系统的数据流连接，一个步骤的输出是下一个步骤的输入。

在管道/过滤器风格的软件体系结构中，每个构件都有一组输入和输出，构件读输入的数据流，经过内部处理，然后产生输出数据流。这个过程通常通过对输入流的变换及增量计算来完成，所以在输入被完全消费之前，输出便产生了。因此，这里的构件被称为过滤器，这种风格的连接件就象是数据流传输的管道，将一个过滤器的输出传到另一过滤器的输入。此风格特别重要的过滤器必须是独立的实体，它不能与其它的过滤器共享数据，而且一个过滤器不知道它上游和下游的标识。一个管道/过滤器网络输出的正确性并不依赖于过滤器进行增量计算过程的顺序。

**特点：**

- 使得软构件具有良好的隐蔽性和高内聚、低耦合的特点；
- 允许设计者将整个系统的输入/输出行为看成是多个过滤器的行为的简单合成；
- 支持软件重用。重要提供适合在两个过滤器之间传送的数据，任何两个过滤器都可被连接起来；
- 系统维护和增强系统性能简单。新的过滤器可以添加到现有系统中来；旧的可以被改进的过滤器替换掉；
- 允许对一些如吞吐量、死锁等属性的分析；
- 支持并行执行。每个过滤器是作为一个单独的任务完成，因此可与其它任务并行执行。

## 5.4 以数据为中心的风格

在以数据为中心的风格中，往往将数据存储构件作为核心，与其他构件紧密结合，形成类似于星型结构的拓扑关系。

### 特点：

- 便于多客户共享大量数据，而不必关心数据是何时产生的、由谁提供的及通过何种途径来提供
- 便于将构件作为知识源添加到系统中来
- 对共享数据结构，不同知识源要达成一致
- 需要同步机制和加锁机制来保证数据的完整性和一致性，增大了系统设计的复杂度

## 5.5 分层 风格

在分层风格中，系统将划分为一个层次结构。每一层都具有高度的内聚性，包含抽象程度一致的各种构件，支持信息隐藏。分层有助于将复杂系统划分为独立的模块，从而简化程序的设计和实现。通过分解，可以将系统功能划分为一些具有明确定义的层，较高层是面对特定问题，较低层具有一般性。每层都为上层提供服务，同时又利用了下层的逻辑功能。在分层体系结构中，每一层只对相邻层可见。层次之间的连接件是协议和过程调用。用以实现各层之间的交互。

### 特点：

- 设计者可以将系统分解为一个增量的步骤序列从而完成复杂的业务逻辑。
- 每一层之多和相邻的上下两层进行交互。
- 只要给相邻层提供相同的接口。
- 并非所有系统都能够按照层次来进行划分。
- 很难找到一种合适和正确的层次划分方法。
- 在传输数据是，需要经过多个层次。
- 多层结构难以调试。

## 5.6 反馈控制环 风格

反馈控制环是一种特定的数据流结构。传统数据流结构是线性的，而控制连续循环过程的体系结构应该是环形的。

反馈控制环系统主要包括以下三个部分：

- 过程，指操纵过程变量的相关机制。
- 数据元素，指连续更新的过程变量，包括输入变量、控制变量、操纵变量和相关参考值。
- 控制器，通过控制规则来修正变量，收集过程的实际状态和目标状态，调节变量以驱动实际状态朝目标状态前进。

### 特点：

- 过程控制是连续的，可以利用各种构件和相关规则来设计反馈控制环系统，实现各种功能。
- 反馈控制环结构能够处理复杂的自适应问题