

体系结构设计学习报告

1.设计过程

设计是一种创造性的过程，它考虑如何实现所有客户的需求；设计所产生的计划也称为设计。

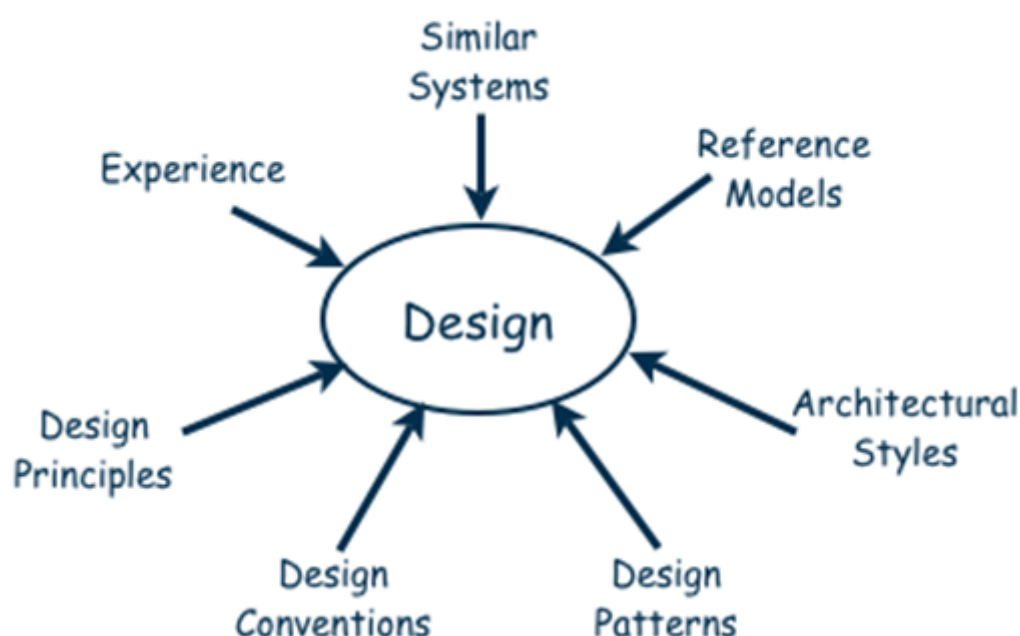
早期的设计专注于系统的体系结构，后续的设计专注于如何实现单个的单元。

设计软件是一种具有智力挑战性的任务。

通过学习优秀的设计例子可以提高设计技巧。

大部分设计工作是例程设计 (routine design), 通过对相似问题的解决方案进行复用和调整来解决当前问题。

多种利用现有解决方案的办法（克隆、参考模型）。



如果我们需要解决的问题没有参考模型，软件体系结构也有一般性的解决方案，称为体系结构风格。专注于一种体系结构风格会产生问题，好的体系结构设计应选择、改进以及集成多种体系结构风格，以产生符合期望的产品。

多种理解决策以及评估选择的体系结构的工具（设计模式、设计公约、创新设计、设计原则）。

软件系统设计是一个迭代的过程，最终结果是软件体系结构文档 (SAD)。

2.分解和视图

创建逐层细化的层次式信息。

一些设计问题没有现成的解决方案，设计人员必须利用分解隔离出系统的关键问题。

流行的设计方法:①面向功能的分解②面向特征的分解③面向数据的分解④面向进程的分解⑤面向事件的分解⑥面向对象的设计

当系统的每个活动都仅由对应的软件单元实现，并且每个软件单元的输入和输出都已经明确地被定义时，设计才可以说是模块化的。如果一个软件单元的接口能够准确无误地指定该单元的外部可见行为，则称该软件单元是定义明确的。

常见体系结构视图:①分解视图②依赖视图③泛华视图④执行视图⑤实现视图⑥部署视图⑦工作分配视图

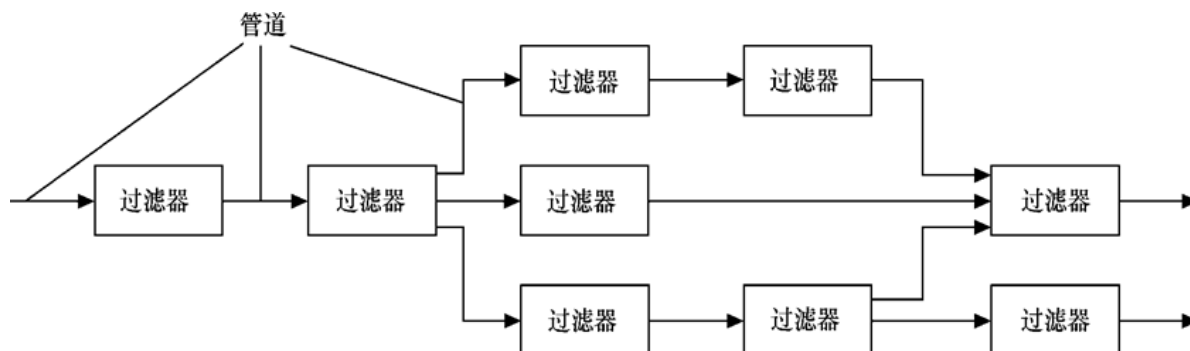
3.体系结构风格

体系结构风格反映了领域中众多系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。对体系结构风格的研究和实践为大粒度的软件复用提供了可能。

经典的体系结构风格：

①数据流风格: 批处理序列; 管道/过滤器。

当输入数据经过一系列的计算和操作构件的变换形成输出数据时，可以应用这种体系结构。管道/过滤器、批处理序列都属于数据流风格。管道/过滤器结构如下图所示。



管道-过滤器风格：把系统任务分成为几个独立的功能模块。这些功能模块采用通过系统的数据流连接。在管道-过滤器风格下，每个功能模块都有一组输入和输出。功能模块称作过滤器（filters）；功能模块间的连接可以看作输入、输出数据流之间的通路，所以称作管道（pipes）。管道-过滤器风格的特性之一在于过滤器的相对独立性，即过滤器独立完成自身功能，相互之间无需进行状态交互

②调用/返回风格：

主程序/子程序；面向对象风格；层次结构。

层次结构风格：整个系统被组织成一个分层结构，每一层为上层提供服务，并作为下一层的客户

③独立构件风格：进程通讯；事件系统。

④虚拟机风格：解释器；基于规则的系统。

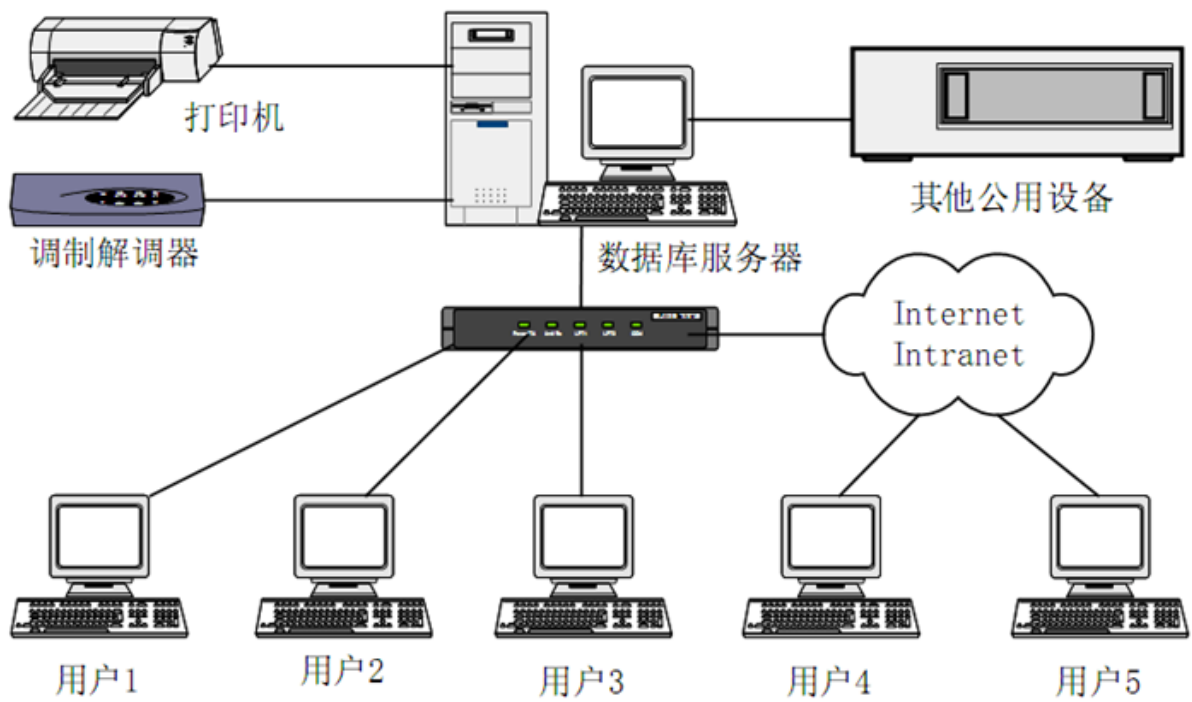
⑤仓库风格：数据库系统；超文本系统；黑板系统。

⑥过程控制环路

⑦C/S风格

客户机/服务器（C/S）体系结构是基于资源不对等，且为实现共享而提出来的。

C/S体系结构有三个主要组成部分：数据库服务器、客户应用程序和网络。



⑧B/S风格

B/S体系结构主要利用不断成熟的WWW浏览器技术，特别是浏览器嵌入的多种脚本语言B/S体系结构有三个主要组成部分：浏览器、Web服务器、数据库服务器。