## 体系结构的应用场景

首先需要承认的是，在软件设计的不同阶段需要不同的设计方法。面对不同粒度，不同复杂度的系统，你需要使用不同的思考框架。

体系结构设计与算法设计就需要不同的思考框架。

在算法设计中，动态规划、贪心算法等这些算法思想是非常重要的思考方式/思考框架，他们在一定程度上能够保证快速得到计算的结果。它们面临的场景更多类似于“快速在某个字符串中快速寻找某个子串”，这样一些小问题。

算法虽然能够攻克粒度较小的问题，但是在构建系统的时候，往往面临的问题是：软件系统规模大且复杂性高。这时对系统的全局结构设计和规划更加重要。比如说，给你一个需求，设计一个图书管理系统，这时候动态规划这些思考框架就显得有些力不从心了。这时候，体系结构设计就应运而生了，体系结构设计是一种构建复杂系统时的思考框架，注意它服务的对象是粗粒度的复杂系统，所以你也不要奢望它能对你的算法设计有太大的帮助。

## 理解软件体系结构

软件体系结构（Software Architecture）包括构成系统的设计元素的描述、 设计元素

之间的交互、 设计元素的组合模式以及在这些模式中的约束。

理解上述抽象概念并不是一件容易的事情，需要一定的工程经验。这里简单谈谈自己的理解。

### 理解的角度1：（集合角度）

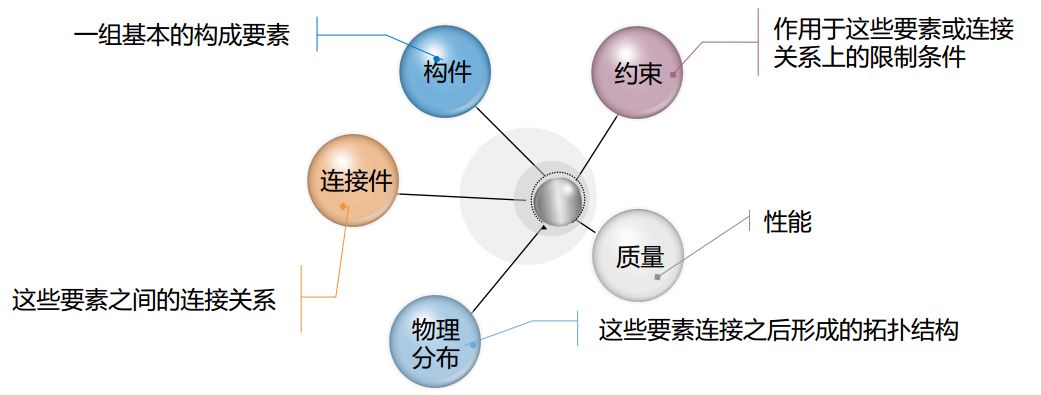
在计算机中，集合是我们一个有利的工具，我们定义事物，理解事物都可以从中获得启发。在集合中，你需要规定基本的元素（这里指构件这些要素），同时还需要制定这些元素的联系方式（这里是指元素的组合模式，及模式中的约束），适当的时候你还需要对具有某些特殊性质的元素进行分类（这里根据元素的作用，将元素分为了两大类：构件 & 连接件）

接下来，你可以从利用常规的集合理论理解该思考框架了。

### 理解的角度2：（实际意义）

我们人类解决复杂问题的方式，基本都是采用分解的方式，即分而治之。所以给定一个系统的时候，我们尝试将其分解成不同的部分，按照一定的划分标准（比如，功能作用），我们可将其分出实际进行处理的模块（构件、连接件），以及负责连接（负责通信）的要素（分布，约束）等。

举个简单的例子，以 “人体”这个系统为例，根据人体各部分的功能，我们可将人体分为不同的构件：头、手、中间部位身体、腿。但是只具备这些要素还是不够的，他们必须连起来，相互配合才能形成真正的人体。



简而言之，软件体系结构 = 构件 + 连接件 + 约束 。

## 体系结构的目标

该体系结构的主要目标是确定影响应用程序结构的需求。布局合理的体系结构可减少与构建技术解决方案相关的业务风险，并在业务和技术要求之间架起一座桥梁。

其他一些目标如下-

公开系统的结构，但隐藏其实现细节。

实现所有用例和方案。

尝试解决各种利益相关者的要求。

处理功能和质量要求。

降低所有权目标并改善组织的市场地位。

改善系统提供的质量和功能。

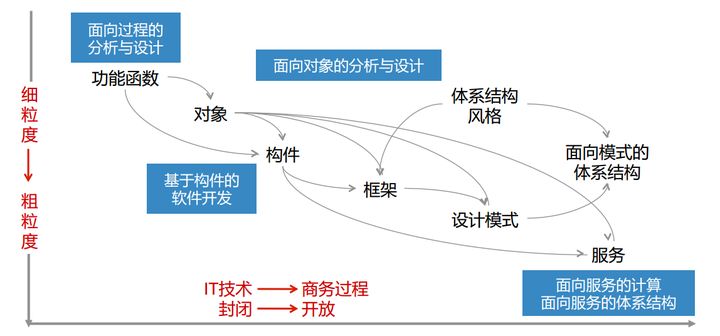
提高外部对组织或系统的信心。

但是软件体系结构仍然是软件工程中的新兴学科。它具有以下限制-

1. 缺乏用于表示架构的工具和标准化方法。
2. 缺乏分析方法来预测体系结构是否会导致实现符合要求的分析方法。
3. 缺乏对体系结构设计对软件开发的重要性的认识。
4. 缺乏对软件架构师角色的了解，以及利益相关者之间的沟通不畅。
5. 缺乏对设计过程的了解，设计经验和设计评估。

## 体系结构的发展

现在软件的复杂性及多变性，导致了软件粒度越来越粗，越来越开放。



## 软件体系结构的建模研究

研究软件体系结构的首要问题是如何表示软件体系结构，即如何对软件体系结构建模。根据建模的侧重点的不同，可以将软件体系结构的模型分为5种：结构模型、框架模型、动态模型、过程模型和功能模型。在这5个模型中，最常用的是结构模型和动态模型。

### 结构模型

这是一个最直观、最普遍的建模方法。这种方法以体系结构的构件、连接件和其他概念来刻画结构，并力图通过结构来反映系统的重要语义内容，包括系统的配置、约束、隐含的假设条件、风格、性质。研究结构模型的核心是体系结构描述语言。

### 框架模型

框架模型与结构模型类似，但它不太侧重描述结构的细节而更侧重于整体的结构。框架模型主要以一些特殊的问题为目标建立只针对和适应该问题的结构。

### 动态模型

动态模型是对结构或框架模型的补充，研究系统的"大颗粒"的行为性质。例如，描述系统的重新配置或演化。动态可能指系统总体结构的配置、建立或拆除通信通道或计算的过程。这类系统常是激励型的。

### 过程模型

过程模型研究构造系统的步骤和过程。因而结构是遵循某些过程脚本的结果。

### 功能模型

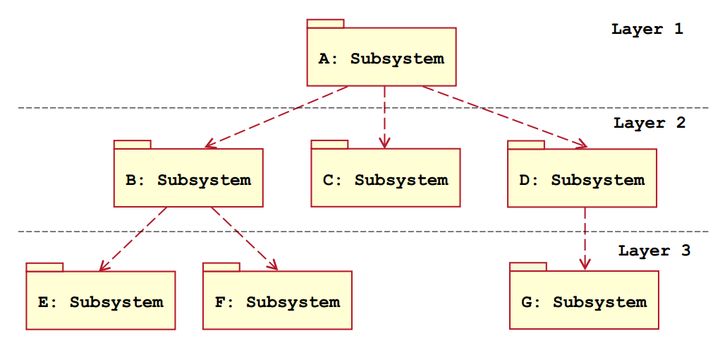
该模型认为体系结构是由一组功能构件按层次组成，下层向上层提供服务。它可以看作是一种特殊的框架模型。

这5种模型各有所长，也许将5种模型有机地统一在一起，形成一个完整的模型来刻画软件体系结构更合适。例如，Kruchten在1995年提出了一个“4+1”的视角模型。“4+1”模型从5个不同的视角包括逻辑视角、过程视角、物理视角、开发视角和场景视角来描述软件体系结构。每一个视角只关心系统的一个侧面，5个视角结合在一起才能够反映系统的软件体系结构的全部内容。

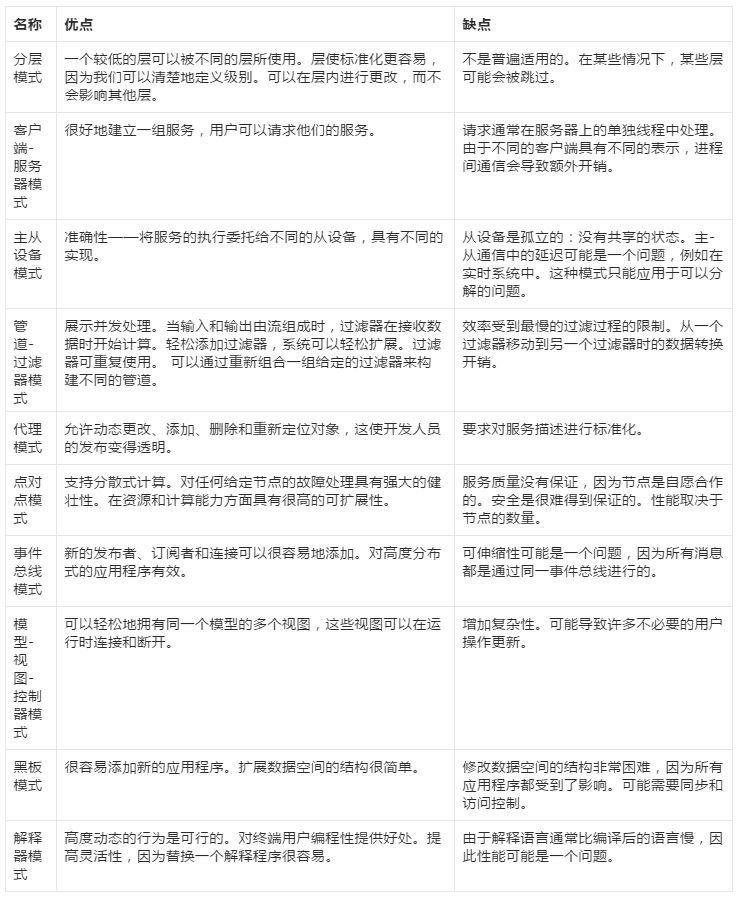
### 层次化

先给出层次化的表现形式：

层次化是模块化的一种特例。在模块化的基础上，加入了一些限制。在层次化结构中，layer可以作为一种特殊的模块（包含特定的子系统集合），在这类模块（layer）的联系中，规定了layer模块只能被设计成类似于双向链表的顺序结构，而不能出现树状结构或者图状结构。



10种常见的体系架构模式优缺点



#### 软件体系结构仍是研究热点

当前，体系结构仍是一个非常新的研究领域，其概念还相当模糊。但软件体系结构作为软件工程领域中的一个组成部分，已经取得了长足的发展，受到大多数软件系统设计和研究人员的重视。

软件体系结构目前较活跃的研究方向包括：（1）软件体系结构形式基础的研究；（2）针对软件体系结构描述中特有的问题研究新的专门的高级语言；（3）建立用于度量和评价软件体系结构的模型和方法；（4）建立面向专门领域的软件体系结构范型库。（5）把软件体系结构从目前的直觉和经验状态过渡到理论。