**定义**

“软件体系结构包括有关软件系统组织的一系列重要决策，包括对构成系统的结构要素及其接口的选择；这些要素之间协作规定的行为；将这些结构和行为要素组成更大的子系统；以及指导该组织的建筑风格。软件体系结构还涉及功能，可用性，弹性，性能，重用性，可理解性，经济和技术约束，折衷和美学方面的考虑。”

软件体系结构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这些模式中的约束。

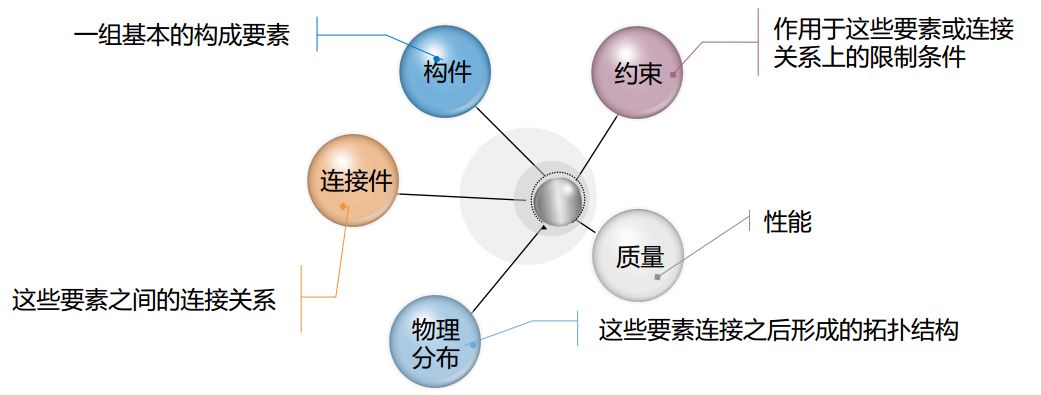
**构件：**一组基本的构成要素。

**连接件：**这些要素之间的连接关系。

**约束：**作用于这些要素或者连接关系上的限制条件。

**质量：**性能。

**物理分布：**这些要素连接之后的拓扑结构。



软件体系结构=构件+连接件+约束

**框架和体系结构的关系**

体系结构的呈现方式是一个设计规约，而框架则是“半成品”的软件。

体系结构的目的是指导软件系统的开发，而框架的目的是设计复用。

**软件体系结构风格**

1. 数据流风格：批处理和管道/过滤器。
2. 调用/返回风格：主程序/子程序、层次结构和C/S。
3. 面向对象风格。
4. 独立部件风格：进程通信和事件驱动。
5. 虚拟机风格：解释器和基于规则的系统。
6. 数据共享风格：数据库系统和黑板系统。

软件体系结构应用现状  
1、**ADL（Architecture Description Language）软件体系结构描述语言**：描述体系结构的概念框架。（C2，Wright，Aesop，Unicon等）  
优秀的ADL的特性：组装性，抽象性，重用性，可配置性，异构性，可分析性等。  
2、**软件体系结构描述构造与表示**：用一定描述方法，对体系结构进行描述。  
eg. “4+1”模型：由逻辑视图，开发视图，过程视图，和物理视图组成。用应用场景将这4个有机结合起来。比较细致地描述了需求和体系结构之间的关系。  
3、**体系结构的设计、分析与验证**：如何将系统分解成相应的组成成分（构件，连接件）  
eg. 风格设计。结构分析，功能分析，非功能分析。  
4、**体系结构的发现、演化与重用**：对已有的系统，提取体系结构等。  
5、**基于体系结构等软件开发**；  
6、**特定领域的体积结构**；  
7、**软件体系结构的支持工具**：软件体系结构的支持工具。  
8、**软件产品线体系结构**：提高重用性。  
9、**建立软件体系结构的方法**；

软件体系结构的研究与应用之中的不足之处：  
1、缺乏统一的软件体系结构概念，导致软件体系结构的研究范围模糊。  
2、ADL繁多，缺乏统一的ADL标准。  
3、软件体系结构研究缺乏统一的理论模型支持。  
4、市面上的多种标准规范或建议标准不易于操作。  
5、软件体系结构性质的研究尚不充分，不能明确给出一个良好的体系极高的属性或评断标准，也没有给出良好体系就高的设计指导原则，因此对于软件开发实践缺乏有力的促进作用。  
6、缺乏有效的支持环境，以及软件体系结构理论研究与环境支持不同步，缺乏有效的体系结构分析、设计、方针和验证工具支持，导致体系结构应用困难。  
7、缺乏有效的体系结构复用方案。  
8、体系结构发现方法研究相对欠缺。