

软件架构与中间件



涂志莹

tzy_hit@hit.edu.cn

哈尔滨工业大学

软件架构与中间件

Software Architecture and Middleware



第1章

系统体系结构的内涵及要点



第1章 系统体系结构的内涵及要点

1.1 系统体系结构概述

1.2 系统体系结构与中间件

1.3 系统体系结构与中间件设计过程

1.1

系统体系结构概述

- 什么是系统体系结构
- 为什么需要“软件系统体系结构”
- 软件系统体系结构的目标与作用
- 软件系统体系结构的发展与演化

什么是系统体系结构

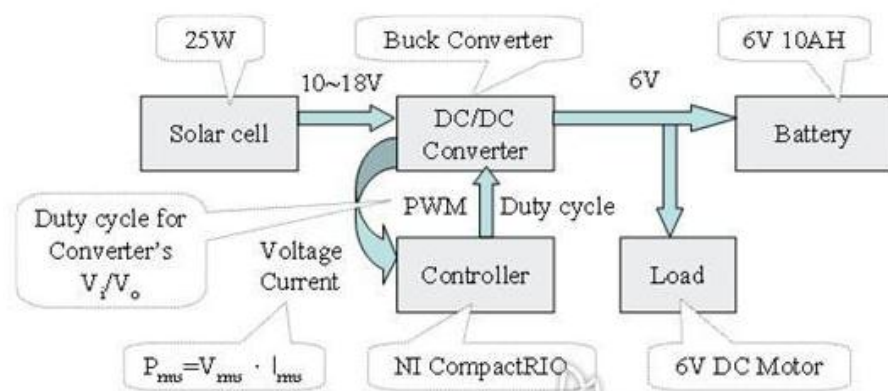
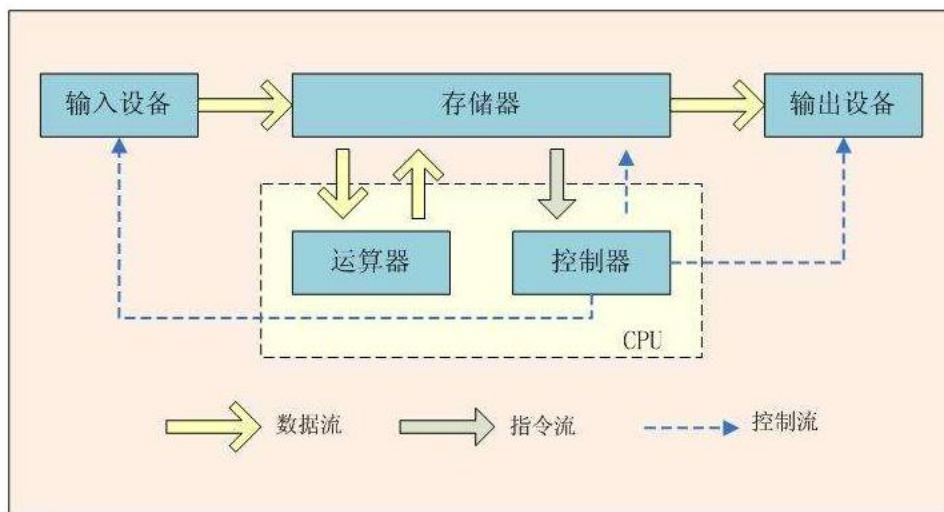
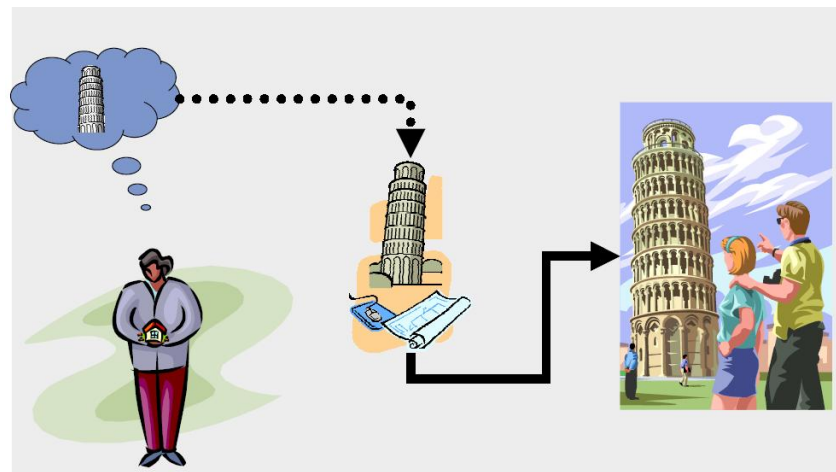
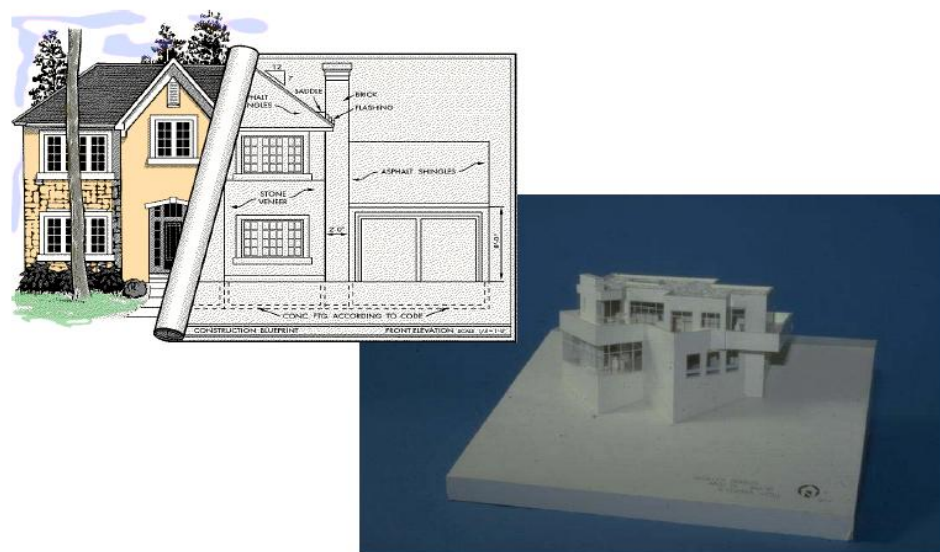
从字面上理解,

软件架构 = 软件的 架构

Software Architecture (SA) = Software's Architecture (S'A)

- 词典里 “架构” 的定义:

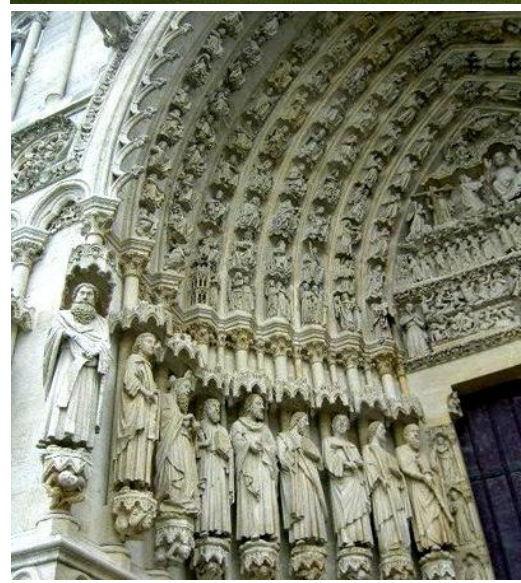
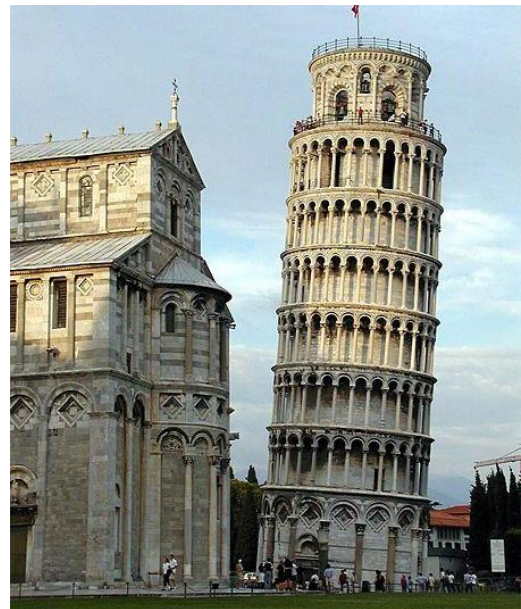
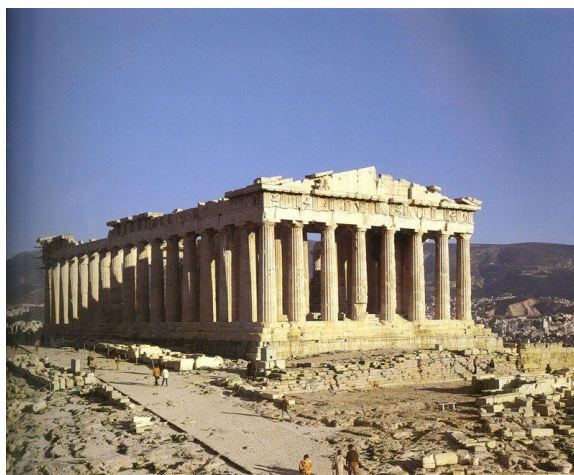
- The art and science of designing and erecting buildings (建筑学: 设计和建造建筑物的艺术与科学);
- A style and method of design and construction (设计及构造的方式和方法);
- Orderly arrangement of parts; structure (部件的有序安排; 结构);
- The overall design or structure of a computer system, including the hardware and the software required to run it, especially the internal structure of the microprocessor (计算机系统的总体设计或结构, 包括其硬件和支持系统运行的软件, 尤其是微处理器内部的结构)。



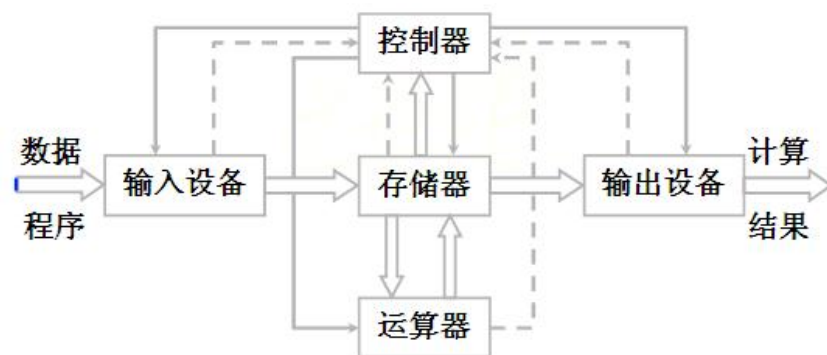
- “架构(Architecture)” 一词起源于建筑学
 - 如何使用基本的建筑模块构造一座完整的建筑？
- 包含两个因素：
 - 基本的建筑模块：砖、瓦、灰、沙、石、预制梁、柱、屋面板...
 - 建筑模块之间的粘接关系：如何把这些“砖、瓦、灰、沙、石、预制梁、柱、屋面板”有机的组合起来形成整体建筑？
- 建筑设计原则
 - 坚固
 - 实用
 - 美观

建筑设计不仅是一门科学，而且是一项艺术！

起源于建筑学的“架构”



- 如何将设备组装起来形成完整的计算机硬件系统？
- 包含两个因素：
 - 基本的硬件模块：控制器、运算器、内存储器、外存储器、输入设备、输出设备、...
 - 硬件模块之间的连接关系：总线
- 计算机架构/体系结构的风格：
 - 以存储程序原理为基础的冯·诺依曼结构
 - 存储系统的层次结构
 - 并行处理机结构
 - 流水线结构
 - 多核CPU
 -



- 共性：

- 一组基本的构成要素——构件
- 这些要素之间的连接关系——连接件
- 这些要素连接之后形成的拓扑结构——物理分布
- 作用于这些要素或连接关系上的限制条件——约束
- 质量——性能

- 软件架构(Software Architecture, SA) :
 - 构件：各种基本的软件构造模块(函数、对象、模式等);
 - 连接件：将它们组合起来形成完整的软件系统;
 - 物理分布
 - 约束
 - 性能

- “There is no standard, universally-accepted definition of the term, for software architecture is a field in its infancy, although its roots run deep in software engineering. ”

- Structural issues include
 - the organization of a system as a composition of components;
 - global control structures;
 - the protocols for communication, synchronization, and data access;
 - the assignment of functionality to design elements;
 - the composition of design elements;
 - physical distribution;
 - scaling and performance;
 - dimensions of evolution;
 - and selection among design alternatives.

SA={components, connectors, constrains}

● Architecture={component, connector, environment, principle}.

- the fundamental organization of a system embodied in its **components**, their **relationships** to each other, and to the **environment**, and the **principles** guiding its design and evolution
- 架构是以构件、构件之间的关系、构件与环境之间的关系以及指导系统设计与演化的原理为内容的某一系统的基本组织结构

- $SA = \{\text{elements, form, rational}\}$

- 软件架构是由一组具有一定形式的元素(elements)构成:
 - 这组元素分成3 类: 处理元素(processing elements)、数据元素(data elements)和连接元素(connecting elements);
 - 处理元素负责对数据进行加工, 数据元素是被加工的信息, 连接元素把架构的不同部分组合连接起来。
- 软件架构形式(form) 是由专有属性(properties)和关系(relationship)组成
 - 专有属性用于限制软件架构元素的选择
 - 关系用于限制软件架构元素组合的拓扑结构
- 在多个架构方案中选择合适的架构方案往往基于一组准则(rational)。

● $SA = \{\text{elements, relations, properties}\}$

- The software architecture of a computing system is the set of structures needed to reason about the system, which comprise software **elements**, **relations** among them, and **properties** of both.

- 软件架构(SA):
 - 提供了一个结构、行为和属性的高级抽象
 - 从一个较高的层次来考虑组成系统的构件、构件之间的连接, 以及由构件与构件交互形成的拓扑结构
 - 这些要素应该满足一定的限制, 遵循一定的设计规则, 能够在一定的环境下进行演化
 - 反映系统开发中具有重要影响的设计决策, 便于各种人员的交流, 反映多种关注, 据此开发的系统能完成系统既定的功能和性能需求

架构 = 构件 + 连接件 + 拓扑结构 + 约束 + 质量

Architecture = Components + Connectors + Topology + Constraints + Performance

为什么要学习 “软件系统体系结构”

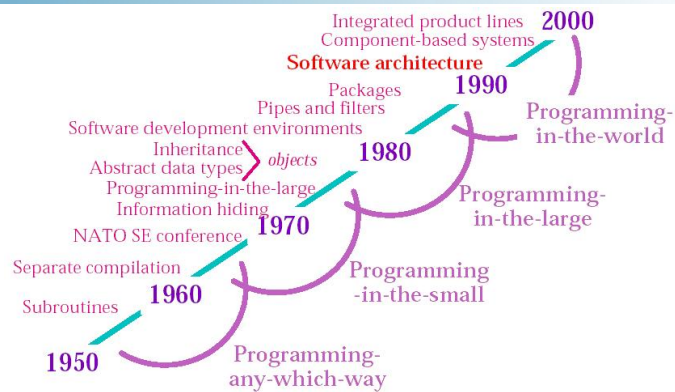
- 最早指出SA的重要性的是大师Edsger Dijkstra(1930-2002)

"..the larger the project, the more essential the structuring!" (1968)



程序规模与软件架构

- 程序 = 算法 + 数据结构 (1960's)
- 程序 = 子程序 + 子程序 (1970's)
- 对象 = 算法 + 数据结构
程序 = 对象 + 对象 (1980's)
- 程序 = 构件 + 连接件 (1990's)



- 随着软件系统规模越来越大、越来越复杂
 - 用户需求(功能性)越来越复杂, 变化越来越频繁;
 - 用户对软件质量(非功能性)的要求也越来越高;
 - 如何将成百上千个功能组合起来, 同时满足用户质量需求, 变得越来越困难。
- 此时, 整个系统的全局结构和设计显得越来越重要。
 - 很多质量需求主要体现在架构中而不是功能模块内部的实现中。
- **结论:** 对于大规模的复杂软件系统来说, 对系统全局结构的设计比起对算法的选择和数据结构的设计明显重要得多。

● 某大型ERP软件

- 软件功能模块>1000个
- 数据表>1000张
- 软件用户>1000人
- 并发用户>50人

● Windows操作系统代码量

- Windows 95: **1500**万行
- Windows 98: **1800**万行
- Windows XP: **3500**万行
- Windows 10: **5000**万行

● 阿里巴巴2018.11.11

- 11日零点刚过2分05秒成交额破100亿元
- 全网交易额2135亿元
- 全天10亿包裹
- 全天实时计算处理峰值达到了每秒17.18亿条

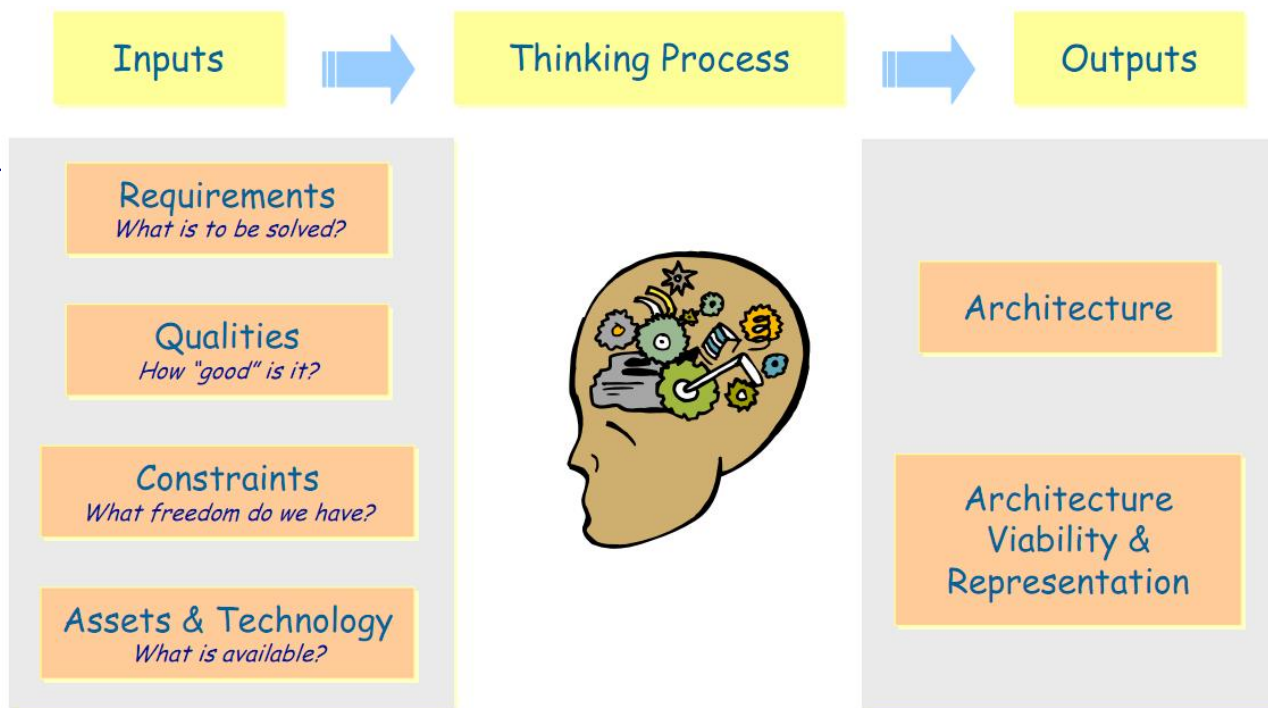
- 构件化设计organization of the system as a composition of components
- 全局控制结构 global control structures
- 通讯协议 protocols for communication
- 异步处理synchronization
- 数据存取data access
- 设计元素的功能分配assignment of functionality to design elements
- 设计元素组合composition of design elements
- 物理部署physical distribution
- 伸缩性与性能scaling and performance
- 演化dimensions of evolution
- 替代方案的选择selection of alternatives
- 等等

- 对于大型复杂软件系统，需要专业的**架构师**来负责系统架构的设计。
- Like a building architect, much of the IT Architect's work is focused on the front end of the **solution life-cycle**.
- The IT Architect may also be involved during the construction of a solution as an advocate for the client, as the **ultimate authority** on the architecture that was produced to address the clients business problem, and to provide **technical leadership** and guidance to the construction team.

——引自IBM资料

对于大型复杂软件系统，需要专业的**架构师**来负责系统架构的设计。

- 是技术领导，可能不是一个人
- 了解开发流程
- 精通业务领域知识
- 精于设计
- 是编程高手
- 优秀的表达能力
- 能够果断决策
- 是谈判专家



- InfoQ <https://infoq.cn>
- 51CTO <http://www.51cto.com/>
- ZDNet <http://www.zdnet.com/>
- 技术大牛博客...
- 开源项目网站...
- 技术公众号...
-

架构师不是教出来的
架构师是实践出来的



第I章

系统体系结构的内涵及要点

Thanks for listening

涂志莹

哈尔滨工业大学计算机学院

企业与服务计算研究中心