

软件体系结构基础

模块一

提纲

- 01 软件发展历史
- 02 软件体系结构定义
- 03 软件体系结构相关概念
- 04 软件体系结构起源
- 05 软件体系结构意义
- 06 软件体系结构最新进展

软件开发历史

- 1970
 - 汇编语言
 - 适用于小型程序
- 1980
 - 高级语言
 - 面向过程理论
 - 数据流/控制流设计方法
- 1995
 - 应用程序开发库 (类库/函数库)
 - 面向对象理论
 - 面向对象建模以及设计技术
- Future
 - 应用程序开发框架: J2EE, .NET
 - 组件技术: COM/DCOM, CORBA ...
 - 对象建模/设计标准化: UML
 - 模型驱动开发: MDA
 - ...

软件开发历史

» 软件开发历史回顾

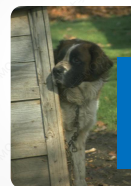
- 软件的规模和复杂度 **越来越高**
软件应用领域: 科学计算、工业生产、商业、教育、娱乐
- 软件的抽象程度 **越来越高**
机器语言 -> 汇编语言 -> 高级语言 -> 开发框架
面向过程编程 -> 面向对象编程 -> 面向切面编程

架构设计的重要性

» 软件开发结果

- 好的架构设计对于软件系统成败至关重要
- 架构设计比数据结构和程序算法更为重要

为什么要做架构设计?



一人就可完成构建
无需建模
过程简单
工具简单

为什么要做架构设计

团队设计更为高效
需要建模
健全的过程
有力的工具



为什么要做架构设计



不同点

- 规模
- 开发团队实力
- 过程
- 所需技术
- 成本
- 利益相关者
- 工作计划
- 风险

软件体系结构定义

The software architecture of a program or computing system is the **structure** or structures of the system, which comprise **software elements**, the externally **visible properties** of those elements, and the **relationships** among them.

----《Software Architecture in Practice》, Addison-Wesley 1997

Architecture is the organizational **structure** of a system. An architecture can be recursively decomposed into **parts** that interact through **interfaces**, **relationships** that connect parts, and **constraints** for **assembling parts**. Parts that interact through interfaces include classes, components and subsystems.

----UML 1.3

软件体系结构定义

Software architecture is the **fundamental organization** of a system, embodied in its **components**, their **relationships** to each other and the environment, and the **principles** governing its design and evolution

----IEEE 1471-2000

软件体系结构定义

» 软件架构

- **软件元素**: 功能、接口、程序、类模块、层、子系统、客户端/服务器等
- **可见属性**: 提供服务、性能特征、错误处理、共享资源使用等
- **关系**: 这些元素之间的组合机制

» 架构是商业和技术决策的结果

软件体系结构定义

软件体系结构组成:

- 构成元素 — 组件
- 交互机制 — 连接器



简而言之:

- 构成系统的组件和组件间的关系或交互机制
- 软件体系结构设计 = 分解 + 组合

软件体系结构定义

分解/组合

- 降低软件设计和构建的复杂度
- 控制软件开发的风险
- 提高组织管理的效率

需要考虑的问题:

- 如何将系统分解为组件?
- 是否拥有所有必须的组件?
- 组件能否进行集成?

软件体系结构定义

软件体系结构定义可参见CMU网站:

<http://www.sei.cmu.edu/architecture/definitions.html>

相关概念

组件

- 系统的逻辑或功能单元

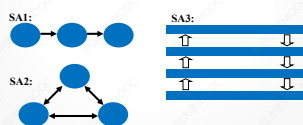
注意:

- 组件可以被拆分为更小的单元或者组件
- 组件提供特定的职责
- 组件是抽象的、概念的词语,在不同的场景中可以是不同的特定对象(例如:模块、子系统、层、包、类等)

相关概念

连接器

- 组件之间的交互规则或机制



相关概念

软件架构的功能属性

- 软件架构中满足功能性需求的属性

软件架构的非功能属性

- 软件架构中满足非功能性需求的属性。如:

- 性能
- 可移植性
- 灵活性/可扩展性
- 可靠性/安全性
- ...

相关概念

» 框架

- 框架是解决特定问题的可重用的应用程序结构。
- 提供解决特定问题的必要的、基本的组件
- 组件间的交互机制和约束
- 提供基于框架开发的应用程序的上下文或环境。
- 通常而言，框架以类库形式提供，例如：.NET 架构、JavaEE 框架等。

Struts 2 架构

Struts



软件体系结构总结

- 架构是系统可看为一个整体的高层次的抽象
- 在架构层次，隐藏所有的实现细节。
- 架构必须支持系统的功能性需求
- 架构必须满足系统的质量属性（非功能性需求）：性能、安全性、可靠性、灵活性、可拓展性等。