



第2章 软件体系结构

2.4 软件体系结构风格 (software architectural style)

刘其成

计算机与控制工程学院

ytliuqc@163.com

2018-9

主要内容

■ 2.4 软件体系结构风格

- 什么是“软件体系结构风格”？
- 为什么要提出“**SA**风格”？
- 典型的**SA**风格都有哪些？

从“建筑风格”开始

- 建筑风格指建筑设计中在内容和外貌方面所反映的特征，主要在于建筑的平面布局、形态构成、艺术处理和手法运用等方面所显示的独创和完美的意境。
- 建筑风格等同于建筑体系结构的一种可分类的模式，通过诸如外形、技术和材料等形态上的特征加以区分。
- 之所以称为“风格”，是因为经过长时间的实践，它们已经被证明具有**良好的工艺可行性、性能与实用性，并可直接用来遵循与模仿(复用)**。

建筑风格

- 建筑风格因受不同时代的政治、社会、经济、建筑材料和建筑技术等的制约以及建筑设计思想、观点和艺术素养等的影响而有所不同。
 - 如外国建筑史中古希腊、古罗马有多立克、爱奥尼克和科林斯等代表性建筑风格；
 - 中古时代有哥特建筑的建筑风格；
 - 文艺复兴后期有运用矫揉奇异手法的巴洛克和纤巧烦琐的洛可可等建筑风格。
 - 我国古代宫殿建筑，其平面严谨对称，主次分明，砖墙木梁架结构，飞檐、斗拱、藻井和雕梁画栋等形成中国特有的建筑风格。

建筑风格

详见微信两篇文章



回到“软件体系结构风格”

- 在软件开发过程中，通常会考虑能否使用重复的体系结构模式，即能否达到体系结构级的软件复用。
- 也就是说，能否在不同的软件系统中，使用同一体系结构。这就是软件体系结构的风格需要研究的问题。

回到“软件体系结构风格”

- 软件系统同建筑一样，也具有若干特定的“风格”；这些风格在实践中被多次设计、应用，已被证明具有良好的性能、可行性和广泛的应用场景，可以被重复使用；实现“软件体系结构级”的复用。
 - 描述一类体系结构
 - 在实践中被多次应用
 - 是若干设计思想的综合
 - 具有已经被熟知的特性，并且可以复用

回到“软件体系结构风格”

- 定义：

- 描述特定领域中软件系统家族的组织方式的惯用模式，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。

- 类比：设计模式 vs 体系结构风格

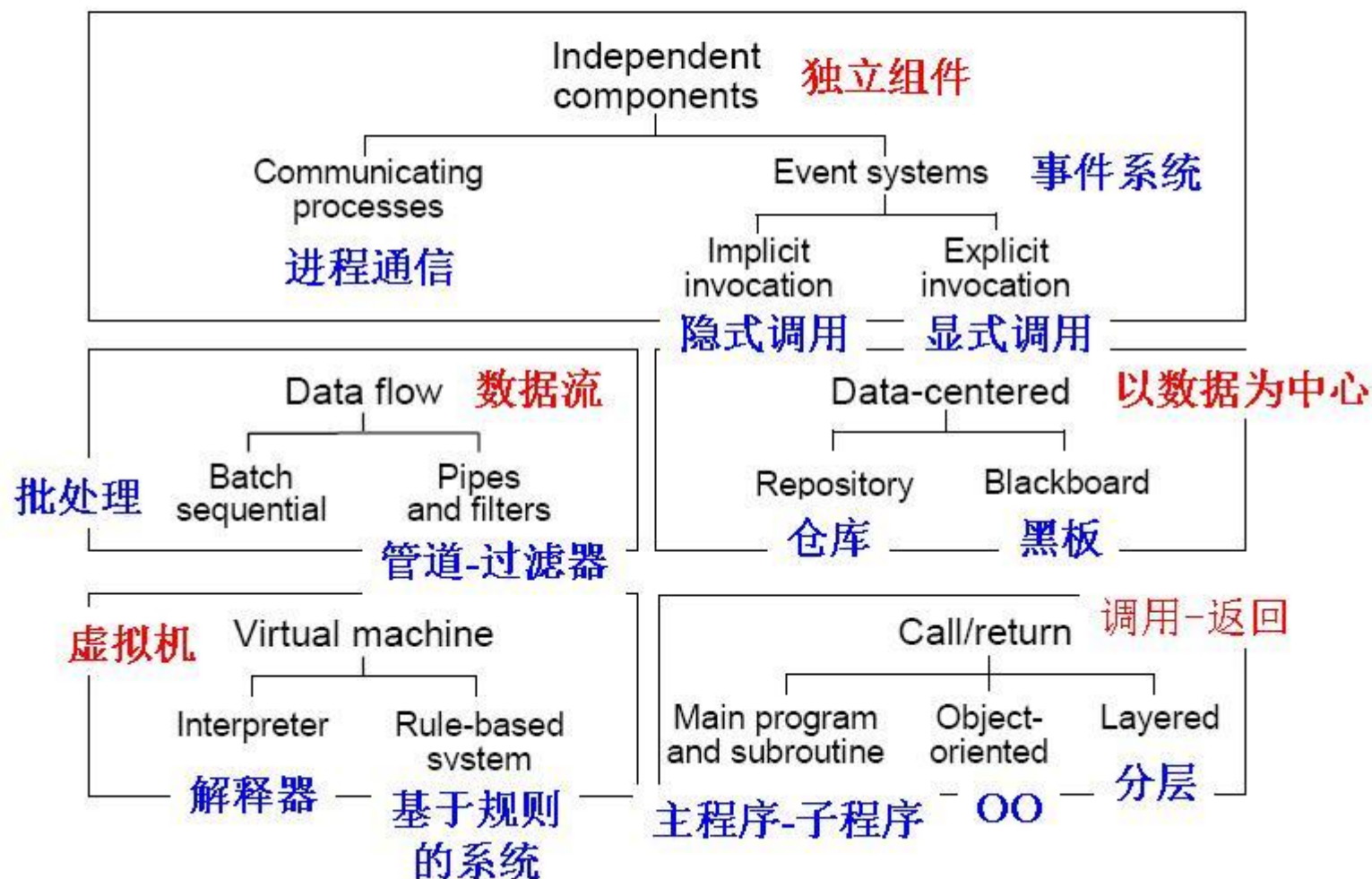
“软件体系结构风格”的组成

- 一组组件类型。例如：数据容器，过程，对象
- 一组连接件类型/交互机制。例如：过程调用，事件，管道
- 这些组件的拓扑分布
- 一组对拓扑和行为的约束。例如：数据容器不能自己存储数据，管道不能是循环的
- 一些对风格的成本和收益的非正式描述。例如：如果你需要重用性并且性能不是很重要，那么可以使用管道风格

软件体系结构风格级的复用

- 软件体系结构风格级的复用，可以使不同的系统可以共享同一个实现代码，一些经过实践证实的解决方案可以可靠地用于解决新的问题。
- 只要系统使用规范的方法来组织，就可以使别的设计者很容易地理解系统的体系结构。
 - 例如，如果把系统描述为“**B/S**”风格，那么不需要给出设计细节也会明白系统是如何组织和工作的。

经典体系结构风格的分类



经典软件体系结构风格

- **Garlan和Shaw**给出的对通用体系结构风格的分类:
 - 数据流风格: 批处理序列; 管道/过滤器
 - 调用/返回风格: 主程序/子程序; 面向对象风格; 层次结构
 - 独立组件风格: 进程通讯; 事件系统
 - 虚拟机风格: 解释器; 基于规则的系统
 - 仓库风格: 数据库系统; 超文本系统; 黑板系统

分类	典型风格
数据流风格 (Data flow)	批处理序列 (Batch sequential), 管道和过滤器 (Pipes and filters)
调用/返回风格 Call/return	主程序/子程序 (Main program and subroutine), 面向对象 (Object-oriented), 层次结构 (Layered)
独立构件风格 (Independent components)	进程通信 (Communicating processes), 事件系统 (Event systems)
虚拟机风格 (Virtual Machine)	解释器 (Interpreter), 基于规则系统 (rule-based system)
仓库风格 (Repository)	数据库系统 (Database system), 黑板系统 (Blackboard system)

基于网络的软件体系结构风格

- 分布式风格(层次结构风格的典型实例):
 - C/S结构
 - B/S结构
- 新型体系结构风格:
 - MVC
 - Cluster
 - SOA
 - Cloud

本课程所要学习的SA风格

- (1) 调用/返回风格: 主程序/子程序; 面向对象; 层次结构
- (2) 数据流风格: 批处理; 管道/过滤器; 过程控制
- (3) 独立组件风格: 基于事件的隐式调用
- (4) 虚拟机风格: 解释器
- (5) 仓库风格: 黑板
- (6) 分布式风格(层次结构风格的典型实例): C/S结构; B/S结构
- (7) 新型体系结构风格: MVC

纯粹的体系结构风格？

- 纯粹的体系风格在实际中很难遇到
- 实际的系统
 - 通常与学术定义相偏离
 - 典型地，融合很多体系风格的特色
- 作为一个架构师，你必须理解“纯”的风格。理解它的优点与缺陷，也要理解背离此种风格之后会带来什么结果

讨论体系结构风格时要回答的问题

- 组件和连接件的类型是什么？
- 可容许的结构模式是什么？
- 基本的计算模型是什么？
- 其使用的常见例子是什么？
- 使用此风格的优缺点是什么？

第二章 思考题

- 1. 软件体系结构的定义
- 2. 组件和连接件的定义
- 3. 设计简单的组件和连接件
 - （相关程序，语言不限。说明程序中的组件和连接件）
 - 例如（1）设计一个子程序，由主程序调用
 - 或者（2）设计多个类和对象，相互调用
 - 或者（3）**JavaBeans**的设计与使用：设计一个求圆的面积的**JavaBean**组件，并说明如何在**JSP**中使用它。
- 4. 简述软件体系结构研究的主要内容
- 5. 软件体系结构风格的定义



谢谢

2018年9月13日