《结构之美(beautiful Architecture spinellis)》作者Spinellis学习报告

**1、基本介绍**  
 本书以实践为基础，通过十二个软件系统的案例，介绍了这些系统的架构设计以及架构设计的过程。本书旨在通过这些案例，帮助读者提高自己的架构设计能力，更好地理解优秀的软件架构应该具备的特征和优势。

**2、软件架构的意义**  
软件设计有设计模式，架构也有架构模式。就是说，构建系统，有一些现成的套路。类似于设计模式，架构在软件开发中为不同的人员提供了共同交流的语言，体现并尝试了系统早期的设计决策，并作为系统设计的抽象，为实现框架和构件的共享和重用、基于架构的软件开发提供了有力的支持。

**3、软件架构分类**  
也叫软件架构风格。软件架构风格是描述某一特定应用领域中系统组织方式的惯用模式。架构风格反映了领域中众多系统所共有的结构和语义特性，并指导如何将各个构件有效地组织成一个完整的系统。

（1）经典架构风格  
1）数据流风格  
如批处理系统，管道等  
2）调用/返回风格  
如主程序/子程序等  
3）独立构件风格  
如事件驱动的系统。  
4）虚拟机风格  
如解释器  
5）仓库风格  
如数据库系统、超文本系统  
（2）层次架构风格  
1）C/S架构  
2）三层C/S架构  
3）B/S架构  
BS架构本质上也算是三层CS架构。  
（3）富互联网应用  
Rich Internet Application，RIA，B/S架构的变种。所谓CS是胖客户端，BS是瘦客户端，说的是CS的客户端交互能力很强，可以做许多工作，BS就不同了，浏览器功能太弱，啥都提交到服务器，极端依赖服务器。但BS发展至今时今日，客户端功能越来越强，体验越来越好，之前的flex，slivelight就不说了，阿贾克斯，以及现在的单页应用，前后端分离，以及浏览器功能的增强，客户端也日益强壮富裕。窃以为，现在的BS程序，就是富互联网应用程序。  
（4）面向服务的架构  
（5）微服务架构

**4、软件架构建模**  
软件架构建模就是如何表达软件架构。根据建模侧重点不同，可以将软件架构的模型分为5忠：结构模型、框架模型、动态模型、过程模型和功能模型。5种模型各有所长，将它们有机统一，形成一个完整的模型来刻画软件架构更合适，比如4+1视图模型；开发视图又叫模块视图；物理视图又叫部署视图；场景是重要系统活动的抽象，它将其余4个视图有机联系起来；对应UML中的用例视图。