Atitit 架构之道 attilax著

[1. 架构的目的是什么？？提高架构可读性。。提高扩展性。。对兼容性也有一定提升。。 4](#_Toc2054)

[1.1. 伸缩性架构设计 4](#_Toc31561)

[1.2. 提升性能架构 4](#_Toc9196)

[1.3. 降低技术难度 提升可维护性架构 4](#_Toc2476)

[1.4. 优化组织架构 4](#_Toc17568)

[1.5. 几乎所有的问题都能通过架构解决 4](#_Toc28055)

[2. 架构方面的概念 4](#_Toc18926)

[2.1.   第2章  架构的种类 4](#_Toc16677)

[2.2. 架构的本质在于其抽象性。它包括两个方面的抽象：业务抽象和技术抽象 4](#_Toc16635)

[2.3. 架构演进 4](#_Toc12924)

[2.4. 组成派”和“决策派” 4](#_Toc26272)

[2.4.1. 组成派 组成派：架构 = 组件 + 交互  5](#_Toc23855)

[2.5. 决策派 5](#_Toc31165)

[3. 发展趋势与历史 主进化路线Cs》》 bs 》3层架构》  SOA》rest》MSA(微服务架构》serverless 6](#_Toc14291)

[3.1. 有哪些架构 6](#_Toc17662)

[4. 做软件架构设计时,根据不同的抽象层次可分为三种不同层次的模式：架构模式(Architectural Pattern)、设计模式(Design Pattern)、代码模式(Coding Pattern)。 6](#_Toc32550)

[5. 其他架构模式 6](#_Toc17382)

[5.1. 语言与架构　第13章　软件架构：面向对象与面向功能 7](#_Toc4977)

[5.2. Hybrid架构 7](#_Toc9503)

[5.3. 分层架构是使用最多的架构模式 Layers模式 也称Tiers模式 7](#_Toc8726)

[5.4. MVC架构 7](#_Toc15456)

[5.5. 微内核架构 •Microkernel（微核）模式 7](#_Toc1123)

[5.6. 元模型架构： 7](#_Toc18535)

[5.7. 管道-过滤器架构： 2.2.2　Pipes and Filters模式 7](#_Toc17953)

[5.8. 点对点（Peer to Peer）对等风格  8](#_Toc21974)

[5.9. 第3章 多层次/多视图软件架构 8](#_Toc6520)

[5.10. 其他架构 8](#_Toc3581)

[6. Cpu架构 x86与arm 9](#_Toc27425)

[7. Os架构 window linux android 9](#_Toc30922)

[8. Atitit 桌面软件gui技术架构发展史attilax总结 9](#_Toc8448)

[9. Atitit 架构的原则attilax总结v2 r19 9](#_Toc13183)

[10. 常见架构方面的书籍 10](#_Toc7823)

[11. 程序设计方法架构 面向过程、面向对象、面向组件、面向服务软件架构 10](#_Toc27106)

[11.1. 第1章 软件架构中面向对象、基于组件、面向代理和面向服务的范式 10](#_Toc17874)

[11.2. 程序设计方法 面向功能 10](#_Toc3573)

[11.3. 1. 面向功能的程序设计方法、结构化程序设计方法、面向数据流的处理方式与结构化分析（Structrued Analysis，即SA）、结构化编程（Structrued Program，即SP）、结构化设计    （Structrued Design，即SD） 10](#_Toc8356)

[11.4. .面向对象方法 11](#_Toc27267)

[11.5. 面向对象、面向服务、面向组件 12](#_Toc32148)

[11.6. 面向模式的软件架构， 12](#_Toc20228)

[12. 开发方法架构   第4章  敏捷软件架构是什么 12](#_Toc25789)

[13. 第10章 细化架构设计 12](#_Toc18337)

[14. 第11章 架构验证 12](#_Toc11318)

[15. 第12章 粗粒度“功能模块”划分 12](#_Toc10853)

[16. 第13章 如何分层 12](#_Toc28932)

[17. 微服务架构与实践 12](#_Toc19158)

[17.1. Line 267: 第 1章 单块架构及其面临的挑战 ............................................. 3 12](#_Toc8384)

[17.2. Line 277: 第 2章 微服务架构综述 .................... 13 13](#_Toc28565)

[17.3. Line 342: 第 8章 日志聚合 .............................. 97 13](#_Toc24909)

[17.4. Line 366: 第 11章 微服务与持续交付 ............ 131 13](#_Toc16817)

[17.5. Line 350: 第 9章 监控与告警 ......................... 105 13](#_Toc20302)

[18. 第14章 架构风格 13](#_Toc27597)

[19. 第15章 使用架构模型 13](#_Toc4414)

[20. 面向模式的软件架构，卷1：模式系统 13](#_Toc3920)

[20.1. 第2章　架构模式 14](#_Toc27736)

[20.2. 第3章　设计模式 14](#_Toc10292)

[20.3. 第8章　模式的发展方向 14](#_Toc12723)

[21. 第10章技术创新 14](#_Toc31793)

[22. 第11章战略路线图 14](#_Toc19767)

[23. 第12章企业执行 14](#_Toc22612)

[24. 组织架构之道 14](#_Toc19808)

[25. 参考资料 14](#_Toc14405)

[25.1. Atitit。软件架构原理与实践概论 14](#_Toc12773)

[25.2. 14](#_Toc3274)

[25.3. 14](#_Toc185)

[25.4. Atitit 常用的软件架构 与 模式 attilax总结.docx 14](#_Toc17222)

[25.5. 14](#_Toc23557)

[25.6. [分享] 软件架构设计之常用架构模式介绍 - chunjuzhong的专栏 - 博客频道 - CSDN.NET.html 14](#_Toc32322)

[25.7. 主要软件类型　适用　的几种典型的　架构模式 - dzldzl - 博客园.html 15](#_Toc26894)

[25.8. 软件架构模式基本概念及三者区别 - jsd2root的博客 - 博客频道 - CSDN.NET.html 15](#_Toc12107)

[25.9. 《面向模式的软件架构，卷1：模式系统》(（德）布施曼　等著)【简介\_书评\_在线阅读】 - 当当图书.html 15](#_Toc5923)

[25.10. 《恰如其分的软件架构(软件架构设计新经典)》(（美）George Fairbanks 著 )【简介\_书评\_在线阅读】 - 当当图书.html 15](#_Toc1187)

[25.11. 15](#_Toc3835)

[25.12. Atitit 桌面软件架构与gui技术发展史attilax总结 v2 r120.docx 15](#_Toc29082)

[25.13. 程序设计方法概述：从面相对象到面向功能到面向对象 - CSDN博客.mhtml 15](#_Toc5692)

[25.14. 《软件架构设计》学习笔记--2--软件架构概念 - aflyeaglenku的博客 - CSDN博客.mhtml 15](#_Toc3719)

[25.15. 学习笔记.\_\_软件架构设计\_\_.第1章.解析软件架构概念 - 技术非技术. [记录酸人, 酸语, 酸事儿.] - ITeye博客.mhtml 15](#_Toc20479)

# 架构的目的是什么？？提高架构可读性。。提高扩展性。。对兼容性也有一定提升。。

## 伸缩性架构设计

## 提升性能架构

## 降低技术难度 提升可维护性架构

## 优化组织架构

## 几乎所有的问题都能通过架构解决

# 架构方面的概念

## 第2章  架构的种类

## 架构的本质在于其抽象性。它包括两个方面的抽象：业务抽象和技术抽象

## 架构演进

## 组成派”和“决策派”

行业大牛对软件架构概念的定义分成两派，应该说相互联系但又互为补充。分别是“组成派”和“决策派”。

可以用如下公式直观表达两种定义：

组成派：架构 = 组件 + 交互   
决策派：架构 = 一组重要决策

### 组成派 组成派：架构 = 组件 + 交互

定义是：”软件系统的架构将系统描述为计算机组件及组件之间的交互“。这里的组件是广泛意义上的，是非常抽象的，既可以是软件也可以硬件，既可以很宏观，也可以很微小。它的描述是以软件本身为描述对象。

## 决策派

定义是软件系统是一些重要方面的决策集合。它包括这样一些问题的重要决策：

* 软件系统的组成。这点上又包含了组合派的定义。
* 选择组成系统的结构元素和他们之间的接口，以及当这些袁术交互协作所体现的行为；
* 如何组合这些元素，使它们逐渐组合成更加大的子系统；
* 用于指导这个系统组织的架构风格；这些元素以及它们的接口、协作和组合。
* 软件架构不仅这种软件本身的结构和行为，还注重其它特性：使用、功能性、弹性、重用、可理解性、经济和技术的限制及权衡，甚至包括美学等。

# 发展趋势与历史 主进化路线Cs》》 bs 》3层架构》  [SOA》rest》MSA(微服务架构](http://www.baidu.com/link?url=A2BxkXZ752i3XmxS3RdircaM30kFmcQGnZHap5ZgiRQWufdnuGq_cj57ATg-l7lg9EMFRHDMsMuusBL6KWLgM_" \t "http://blog.csdn.net/attilax/article/details/_blank)》serverless

## 有哪些架构

# 做软件架构设计时,根据不同的抽象层次可分为三种不同层次的模式：架构模式(Architectural Pattern)、设计模式(Design Pattern)、代码模式(Coding Pattern)。

架构模式是一个系统的高层次策略，涉及到大尺度的组件以及整体性质和力学。架构模式的好坏可以影响到总体布局和框架性结构。

设计模式是中等尺度的结构策略。这些中等尺度的结构实现了一些大尺度组件的行为和它们之间的关系。模式的好坏不会影响到系统的总体布局和总体框架。设计模式定义出子系统或组件的微观结构。

代码模式（或成例）是特定的范例和与特定语言有关的编程技巧。

# 其他架构模式

## 语言与架构　第13章　软件架构：面向对象与面向功能

## Hybrid架构

## 分层架构是使用最多的架构模式 Layers模式 也称Tiers模式

## MVC架构

## 微内核架构 •Microkernel（微核）模式

## **元模型架构**：

元模型架构就是有元数据支撑的架构，现在使用的也很广泛，比如：ORM,.Net 类的设计等都是元数据支持的。元数据有自我描述性比如ORM会描述类对应**[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "http://blog.csdn.net/chunjuzhong/article/details/_blank)**中的表属性对应数据库里的字段，还有IOC类中的引用需要注入哪个类等等都会通过元数据的形式实现。IOC框架通过解析元数据信息使注入和被注入类只通过接口依赖，这样替换注入类很方便。元数据架构是很灵活的架构，可发展空间非常大，元数据架构会经常用反射技术或者动态代码生成技术。我之前做了一个ORM就是用到的元数据架构，我还想给ORM添加依赖注入面向切面编程等特性都很方便的。

## **管道-过滤器架构**： 2.2.2　Pipes and Filters模式

1. 这个模式就像是工厂的流水线，生产原料通过流水线经过很多环节进行处理变成产品。软件也是一样的，网络OSI7层就是消息通过管道内部的很多步处理对消息进行加工过滤转换。再举一个例子，两家企业需要信息交换，但是企业的信息格式和描述规则都不相同，如果想达到交换必须经过处理，所以我们就得用管道过滤器模式，通过管道过滤器模式信息进入管道我们会在管道里添加各种处理功能，比如：数据验证，信息加密，信息解密，信息压缩，信息解压缩，格式转换等功能，对消息进行处理以符合我们要求的消息格式，而且如果需要添加一个新的处理只要把处理的功能插入到管道中即可，这样达到最大的灵活性。应用此模式的有：ASP**[.Net](http://lib.csdn.net/base/dotnet" \o ".NET知识库" \t "http://blog.csdn.net/chunjuzhong/article/details/_blank)**请求模型，Spring 对象构造，Structs 数据请求等。

## 点对点（Peer to Peer）对等风格

## 第3章 多层次/多视图软件架构

## 其他架构

2.2.3　Blackboard模式

SSH架构 ssm架构 ，java net lamp架构 大泥球风格   批量顺序处理风格  分发-订阅风格  map-reduce风格  orm ioc

# Cpu架构 x86与arm

# Os架构 window linux android

# Atitit 桌面软件gui技术架构发展史attilax总结

1.1. 第一阶段 像素级自绘阶段 1

1.2. 第2阶段Native控件阶段 1

1.3. 第3阶段 浏览器plugin阶段 （Sliverlight，flash） 1

1.4. 第4阶段 dsl h5 gui阶段 1

1.5. 第5阶段 浏览器 app模式+ h5 2

# Atitit 架构的原则attilax总结v2 r19

1.1. 稳定原则：优先使用成熟三俩年以后的技术，最新的技术慎重考虑 1

1.2. Rule of three称为"三次原则"，指的是当某个功能第三次出现时，才进行"抽象化"。是DRY原则和YAGNI原则的折中 2

1.3. 面向人类编程（可读性优先），优先于面向机器（性能） 2

1.4. 归一化原则，一切都是文件，一切都是sprite 2

1.5. 高层抽象原则，优先于具体实现，框架 类库优先于自己实现 2

1.6. 移动小屏幕优先原则 2

1.7. 点击触摸 优先于键盘 遥控操作 2

1.8. 适当超前设计，超前抽象一层即可 2

1.9. 超前架构设计，一定要满足未来十年的规划为好 2

1.10. 最小化耦合原则与最大内聚5 2

2. 更抽象的原则，通用语软硬件，与宇宙 2

2.1. 事不过三原则：： 3

2.2. 分而治之 3

3. Qa集合 3

3.1. Business is the king没错，But, business is not the everything。 3

# 常见架构方面的书籍

1. 架构之美 作者:（美）斯宾耐立思，（美）郭西奥斯 1

2. 软件架构设计（第2版）——程序员向架构师转型必备(详解软件架构设计实践过程及方法，要成为软件架构设计师必读书) 2

2.1. 从入门到精通必读经典教程全新升级《C++ Primer Plus(第6版)中文版》>> ?? 2

3. 微服务架构与实践 3

4. 程序员必读之软件架构 4

5. 大数据时代的IT架构设计 7

6. 恰如其分的软件架构 作者:（美）George Fairbanks 著 8

7. 面向模式的软件架构，卷1：模式系统 9

8. 企业级Java EE架构设计精深实践 9

9. 信息架构：超越Web设计（第4版）（全彩） 10

10. 软件架构师的12项修炼：技术技能篇 10

11. 软件架构 （法）穆拉德·沙巴纳·奥萨拉赫 11

12. 企业应用架构模式作者:（英）福勒 12

# 程序设计方法架构 面向过程、面向对象、面向组件、面向服务软件架构

## 第1章 软件架构中面向对象、基于组件、面向代理和面向服务的范式

## 程序设计方法 面向功能

## 1. 面向功能的程序设计方法、结构化程序设计方法、面向数据流的处理方式与结构化分析（Structrued Analysis，即SA）、结构化编程（Structrued Program，即SP）、结构化设计    （Structrued Design，即SD）

## .面向对象方法

的基本特征与几个功能  
  
        面向对象是一种更优秀的的程序设计方法，它的基本思想是使用类、对象、继承、封装，消息等基本概念来进行程序设计。它是从现实世界中客观存在的事物（即对象）出发来构造软件系统，并在系统构造中尽可能运用人类的自然思维方式，强调直接以现实世界中的事物（即对象）为中心来思考问题，认

## [面向对象、面向服务、面向组件](https://www.baidu.com/link?url=ruDERqaG7j0vmQWZ-sx2hrRZ7Bb40d0T9l7qjszETISAaXLbaZOAZWDMi31dTXnMQRBQCDBLl2XHV9JGOBmXQq&wd=&eqid=9e9995520002f98d000000035a59884c" \t "https://www.baidu.com/_blank)

## 面向模式的软件架构，

# 开发方法架构   第4章  敏捷软件架构是什么

# 第10章 细化架构设计

# 第11章 架构验证

# 第12章 粗粒度“功能模块”划分

# 第13章 如何分层

# 微服务架构与实践

## Line 267: 第 1章 单块架构及其面临的挑战 ............................................. 3

## Line 277: 第 2章 微服务架构综述 .................... 13

## Line 342: 第 8章 日志聚合 .............................. 97

## Line 366: 第 11章 微服务与持续交付 ............ 131

## Line 350: 第 9章 监控与告警 ......................... 105

# 第14章 架构风格

# 第15章 使用架构模型

# 面向模式的软件架构，卷1：模式系统

第1章　模式

## 第2章　架构模式

## 第3章　设计模式

## 第8章　模式的发展方向

# 第10章技术创新

# 第11章战略路线图

# 第12章企业执行

# 组织架构之道

# 参考资料

## Atitit。软件架构原理与实践概论

## 

## 

## Atitit 常用的软件架构 与 模式 attilax总结.docx

## 

## [分享] 软件架构设计之常用架构模式介绍 - chunjuzhong的专栏 - 博客频道 - CSDN.NET.html

## 主要软件类型　适用　的几种典型的　架构模式 - dzldzl - 博客园.html

## 软件架构模式基本概念及三者区别 - jsd2root的博客 - 博客频道 - CSDN.NET.html

## 《面向模式的软件架构，卷1：模式系统》(（德）布施曼　等著)【简介\_书评\_在线阅读】 - 当当图书.html

## 《恰如其分的软件架构(软件架构设计新经典)》(（美）George Fairbanks 著 )【简介\_书评\_在线阅读】 - 当当图书.html

## 

## Atitit 桌面软件架构与gui技术发展史attilax总结 v2 r120.docx

## 程序设计方法概述：从面相对象到面向功能到面向对象 - CSDN博客.mhtml

## 《软件架构设计》学习笔记--2--软件架构概念 - aflyeaglenku的博客 - CSDN博客.mhtml

## 学习笔记.\_\_软件架构设计\_\_.第1章.解析软件架构概念 - 技术非技术. [记录酸人, 酸语, 酸事儿.] - ITeye博客.mhtml

作者:: 绰号:老哇的爪子claw of Eagle 偶像破坏者Iconoclast image-smasher

捕鸟王"Bird Catcher kok 虔诚者Pious 宗教信仰捍卫者 Defender Of the Faith. 卡拉卡拉红斗篷 Caracalla red cloak 万兽之王 纵火者

简称：： st Emir Attilax Akbar 圣 埃米尔 阿提拉克斯 阿克巴

全名：：st Emir Attilax Akbar bin Mahmud bin attila bin Solomon bin adam Al Rapanui 圣 埃米尔 阿提拉克斯 阿克巴 本 马哈茂德 本 阿提拉 本 所罗门 本亚当 阿尔 拉帕努伊

常用名：艾提拉（艾龙）， EMAIL:1466519819@qq.com

头衔：

|  |  |
| --- | --- |
| uke | Emir Uke部落首席大酋长，ati协会创始人  uke总部o2o负责人，全球网格化项目创始人，  圣阿提拉克斯国王 |
| 科技领域 | UTSC uke技术标准化委员会委员长 uke 首席cto 软件部门总监 技术部副总监 研发部门总监主管 产品部副经理 项目部副经理 uke科技研究院院长 uke软件培训大师  Ati组织科研研究院创始人 |
| 文艺领域 | ， ，， uke机车协会主任 uke纹身协会  uke交友协会会长 uke捕猎协会会长  Ati文艺协会会长 ati文学协会 |
| 行政领域 | Gchsp总裁 gchsp常委 GsP创始人 |
| 媒体传播领域 | uke出版社编辑总编 宣传布道总策划  Ati传媒总部 |
| 渔猎军事领域 | uke保安部首席大队长  Uke 户外运动协会理事长 度假村首席大村长  Ati打猎协会 |
| 法学 | 法学研究会 制度研究会 |
| 管理领域 | 工商管理学 公共管理与社会服务  ,uke制度检查委员会副会长 |
| 教育领域 | uec学院校长， uecip图像处理机器视觉专业系主任 uke文档检索专业系主任  Uke图像处理与机器视觉学院首席院长  uke终身教育学校副校长  靓号研究院 |
| 经济领域 | uke波利尼西亚区大区连锁负责人 汤加王国区域负责人 uke克尔格伦群岛区连锁负责人，莱恩群岛区连锁负责人，uke布维岛和南乔治亚和南桑威奇群岛大区连锁负责人  Uke软件标准化协会理事长理事长 Uke 数据库与存储标准化协会副会长  直达巴士西北区负责人 直达巴士长沙与西安分部部长  润昌通讯软件事业部总裁 执行长 分部负责人 执行委员会主席  Ati经济研究所 |
| 历史领域 | 历史事业部 ati历史研究院 |
| 社会科学领域 | 社科学院 ati文化部 |
| 自然科学领域 | Uke研究院院长兼首席研究员 科学家  Ati自然科学研究院 |
| 宗教神学领域 | uke宗教与文化融合事务部部长 大师master  uke制度与重大会议委员会委员长 ati宗教事务所 |
| 医学领域 | Uke医院 与医学院方面的创始人 |

转载请注明来源：attilax的专栏 <http://blog.csdn.net/attilax>

<http://www.cnblogs.com/attilax/>

Microblog

<http://weibo.com/u/5941179815> (common attilax)

<https://weibo.com/p/1005055941179815> （attilax201707,bek weibo）

<http://weibo.com/u/5487832265> (tech,for blog auto gene)

知乎空间

https://www.zhihu.com/people/ati-att/activities

Qq 1466519819 小号112237553

微信attilax 小号attilax201708

微博 attilax2016 小号attilax201707

--Atiend v19