Atitit.软件架构高扩展性and兼容性原理与概论实践attilax总结

[1. 什么是可扩展的应用程序？ 1](#_Toc12137)

[2. 松耦合(ioc) 2](#_Toc5743)

[3. 接口的思考 2](#_Toc17515)

[4. 单一用途&模块化，小粒度化 2](#_Toc24768)

[5. 组合（Composition），而不是继承（inheritance） 2](#_Toc27286)

[6. Ocp原则开闭原则 2](#_Toc31619)

[7. Plugin系统 2](#_Toc7330)

[8. 流程扩展工作流系统，流程自定义 2](#_Toc24577)

[9. Ui扩展 html5 3](#_Toc31322)

[10. 数据独立性 3](#_Toc13282)

[11. 脚本与hotdeploy 3](#_Toc3012)

[12. 表处理扩展if else (数据与数据处理相互分离） 3](#_Toc23583)

[13. 系统被扩展的几种形式（方法级别，模块级别） 3](#_Toc29374)

[14. 常用软件扩展机制 3](#_Toc10434)

[14.1. Ff的扩展机制》》》》插件体系 4](#_Toc9792)

[14.2. 配置化 4](#_Toc26515)

[14.3. dsl 4](#_Toc27873)

[15. 面临的扩展性与兼容性问题 4](#_Toc23829)

[15.1. 不同的项目与不同的实例启动 4](#_Toc14418)

[15.2. 不同的项目与实例配置文件 4](#_Toc24193)

[15.3. Web.xml怎么办？？ 4](#_Toc20974)

[15.4. 跳转到同一功能spec的不同实现 4](#_Toc23725)

[15.5. 不同项目的同一功能就实现可以放在同一上级模块package下 5](#_Toc2191)

[16. 架构即未来：现代企业可扩展的Web架构、流程和组织(原书第2版) 5](#_Toc28406)

[17. 高扩展性网站的50条原则(网站运营必备宝典)作者:[美]Martin L. Abbott　Michael T. Fisher 著 6](#_Toc21067)

# 什么是可扩展的应用程序？

一个可扩展的应用程序应该能够以某种方式实现增长，并且添加、删除、增强、重构某些组件，对于其他组件的影响微乎其微

当出现下面的这些情况时，说明元素已经具有可扩展性了：

* 该元素可以很容易地与其他元素进行互换，而不会破坏应用程序
* 该元素可以轻松重用于项目外部
* 该元素可以成功地进行单元测试

# [松耦合](http://en.wikipedia.org/wiki/Loose_coupling)(ioc)

soma.js中提供了一系列用于架构解耦和测试的工具，以及各种设计模式解决方案，比如[依赖注入](http://en.wikipedia.org/wiki/Dependency_injection)（dependency injection）、[观察者模式](http://en.wikipedia.org/wiki/Observer_pattern)（observer pattern）、[中介者模式](http://en.wikipedia.org/wiki/Mediator_pattern)（mediator pattern）、[外观模式](http://en.wikipedia.org/wiki/Facade_pattern)（facade pattern）、[命令模式](http://en.wikipedia.org/wiki/Command_pattern)（command pattern），[面向对象](http://en.wikipedia.org/wiki/Object-oriented_programming)（OOP）工具集，并提供了一个DOM操作模板引擎作为可选插件。

作者:: 绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ）

汉字名：艾提拉（艾龙），   EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

# 接口的思考

# 单一用途&模块化，小粒度化

粒度更小，更容易扩展

# 组合（Composition），而不是继承（inheritance）

# Ocp原则开闭原则

# Plugin系统

# 流程扩展工作流系统，流程自定义

# Ui扩展 html5

# 数据独立性

每个功能使用不同的独立数据，这样可以提升扩展性。。比如订单处理。。接受订单，处理订单\*update ，使用不同的表 ，就可以互相不影响的功能修改以及表结构修改。。

# 脚本与hotdeploy

# 表处理扩展if else (数据与数据处理相互分离）

# 系统被扩展的几种形式（方法级别，模块级别）

1. ****运行时对象装配。****
2. ****5）基于模块的运行时动态扩展。**** 到现在为止，我们讨论的都是比较细粒度的在技术上的扩展。想象一下，如果增加一些功能，就去继承各种各样的类，然后经过复杂的装配过程，才能完成一个功能的添加和改进。即使你新增加了类，改动了Spring的配置文件，实现新的装配，但仍然有一点，你需要编译整个项目。也就是说，前四种都没有达到模块级别的扩展。  
         OSGi给出了解决方案，可以实现模块级别的动态扩展，而且是运行时的。所谓运行时模块的动态扩展，比如说你需要增加一些新的功能，你可以将新开发的类和文件按照Bundle进行组织，然后直接扔到运行时环境下，这些功能就可以用了。
3. ****6）基于中间语言。**** 框架通常定义了一些Hotspot（热点），在这些点上，可以进行扩展。平台，可扩展性是最强的，比如window 平台，你可以开发应用程序运行在上面。windows提供了几千个API，你可以使用他们来开发应用程序

由此推断，从技术上来说，要获得最大的可扩展性，就要通过一种或多种中间语言来进行扩展。比如现在流行的OpenSocial API和Facebook都是这个思路。这样说来，DSL的出现就是顺理成章的了，它是为了满足在某个业务领域的扩展而设计的

# 常用软件扩展机制

## Ff的扩展机制》》》》插件体系

## 配置化

## dsl

# 面临的扩展性与兼容性问题

## 不同的项目与不同的实例启动

通过不同的bat启动文件来实现。。

可以通过启动脚本来设置不同的实例

## 不同的项目与实例配置文件

参照mysql，通过制定配置文件来实现

## Web.xml怎么办？？

暂时没办法，可以通过jetty等可编程中间件来实现。。

## 跳转到同一功能spec的不同实现

比如游戏和点播系统的用户中心，ui界面就不同。。

可以使用查表法。。使用js，根据项目的id来跳转不同的功能ui加载，项目id在启动脚本里面传入。。 同一项目的不同实例只能跳转到相同功能，如果需要不同，在加个实例id判断吧。。

Prjid,instan\_id,fun\_implet

## 不同项目的同一功能就实现可以放在同一上级模块package下

重名问题，通过添加后缀\_prj 来解决。。

如果项目很多，应该通过前缀来解决，这样好找。。

# 架构即未来：现代企业可扩展的Web架构、流程和组织(原书第2版)

第一部分　可扩展性组织的人员配置

第1章　人员和领导力对扩展性的影响 …… 2

第2章　可扩展性技术组织的角色 …… 17

第3章　组织的设置 …… 41

第4章　领导力秘籍 …… 80

第5章　管理秘籍 …… 116

第6章　关系、思维和商业案例 …… 141

第二部分　构建可扩展的过程

第7章　过程是可扩展的关键 …… 154

第8章　管理故障和问题 …… 169

第9章　危机管理和升级 …… 189

第10章　生产环境的变更管理 …… 210

第11章　确定应用发展的预留空间 …… 233

第12章　确立架构原则 …… 248

第13章　联合架构设计和架构审查委员会 …… 267

第14章　敏捷架构设计 …… 284

第15章　聚焦核心竞争力：自建与外购 …… 296

第16章　确定风险 …… 310

第17章　性能与压力测试 …… 328

第18章　障碍条件与回滚 …… 351

第三部分　可扩展的架构方案

第19章　构建故障隔离的架构 …… 368

第20章　AKF扩展立方体介绍 …… 389

第21章　为扩展分割应用 …… 404

第22章　为扩展分割数据库 …… 426

第23章　为扩展而缓存 …… 452

第24章　为扩展而异步 …… 472

第四部分　其他的问题和挑战

第25章　海量数据 …… 492

第26章　云计算的突飞猛进 …… 517

第27章　云计算准备就绪 …… 550

第28章　应用监控 …… 564

第29章　规划数据中心 …… 581

第30章　纵观全局 …… 608

# 高扩展性网站的50条原则(网站运营必备宝典)作者:[美][Martin](http://www.dangdang.com/author/Martin_1" \t "http://product.dangdang.com/_blank) L. [Abbott](http://www.dangdang.com/author/Abbott_1" \t "http://product.dangdang.com/_blank)　[Michael](http://www.dangdang.com/author/Michael_1" \t "http://product.dangdang.com/_blank) T. [Fisher](http://www.dangdang.com/author/Fisher_1" \t "http://product.dangdang.com/_blank) 著

第1章　化简方程　1

第2章　分布工作　23

第3章　横向扩展设计　35

第4章　使用正确的工具　51

第5章　不要重复工作　67

第6章　积极利用缓存　81

第7章　从错误中吸取教训　109

第8章　数据库原则　125

第9章　容错设计与故障控制　141

第10章　避免或分发状态　159

第11章　异步通信和消息总线　171

第12章　其他原则　183

第13章　原则回顾和优先级划分　203

参考资料

如何一步一步打造高可扩展性的应用程序？-CSDN.NET.html

软件可扩展性：来自星巴克的经验 - Java,java框架 - language - ITeye论坛.html

可扩展性（Extensibility）——构建灵活系统的思考 - me.think(everything.about(\_software\_)).serialize(this);\_\_古路刀客 - ITeye技术网站.html

架构师不可不知的十大可扩展架构 - 51CTO.COM.html（ imp ，还没看完）

atiend