Atitit 团队工具链体系打造---开发方面提升效率的软件与技术体系 attilax总结.docx

[1. 编程语言 2](#_Toc15276)

**[1.1. 编程语言的发展 从机器语言，汇编语言到本地native语言(c c++)到vm语言(java  c#) 再到脚本语言(js php python等) 再到dsl](#_Toc501)** [2](#_Toc501)

**[1.2. 1.4. 混合编程 通过在高层语言嵌入dsl](#_Toc11674)** [2](#_Toc11674)

[1.3. 业务逻辑的dsl 目前还没流行，有待于未来的发展 2](#_Toc19421)

[1.4. 目前语言排行榜 3](#_Toc30574)

[2. Gui界面开发 3](#_Toc25564)

[2.1. Gui的发展历程与未来趋势 3](#_Toc2724)

[2.2. 客户端mvc 优先于服务端mvc 3](#_Toc26055)

[2.3. 客户端ui 优先于服务端ui 4](#_Toc27352)

[2.4. 适当的全栈 4](#_Toc18791)

[3. archi架构 4](#_Toc4328)

**[3.1.](#_Toc19336)** [常见架构发展历程（按时间](#_Toc19336) **[主进化路线Cs》》 bs 》3层架构》 SOA》rest》MSA(微服务架构](#_Toc19336)** [4](#_Toc19336)

[3.2. 项目结构 为了方便，尽可能优先使用模块》service/dao结构 4](#_Toc30766)

[3.3. Hybrid混合框架模式 5](#_Toc27864)

[3.4. 中间端ioc的实现spring guice ServiceLoctor  ，解决spring启动慢的问题 5](#_Toc11891)

[3.5. 架构方面的提升 适当的简化架构 5](#_Toc7304)

[4. 数据存储 5](#_Toc17544)

[4.1. 数据存储orm方面 apache dbutil》mybatis》hibernate 5](#_Toc5373)

[4.2. 适当使用sp存储过程，触发器，视图。。大力简化开发 6](#_Toc11438)

[4.3. Sql hql vs 程序代码 6](#_Toc25374)

[5. dev开发 6](#_Toc15665)

[5.1. 热部署对提升开发效果很有用 tomcat+ jrebel，或者resin。。 6](#_Toc5378)

[5.2. 任务栏拉到左边，可以给宝贵的上下空间挪出一点地方 7](#_Toc225)

[5.3. Ide黑色背景主题不错 7](#_Toc13059)

[5.4. 通用的轻量级编辑器，有语法提示notepad++ ，editplus Sublime 等 7](#_Toc3150)

[5.5. 其他，还有待于搜集 7](#_Toc32393)

[6. 硬件 8](#_Toc1525)

[6.1. 大屏幕 ssd 多核cpu 8](#_Toc3079)

[6.2. 参考资料 8](#_Toc16576)

# 编程语言

## **编程语言的发展 从机器语言，汇编语言到本地****native语言(c c++)到vm语言(java  c#) 再到脚本语言(js php python等) 再到dsl**

语言层次越高，可读性一般越好，可移植性越好，不过性能也越差了。。当然如果使用同样的类库的话，只是写点胶水代码的话，差别到是相对来说不大了。

机器语言， 可以说彻底淘汰，唯一的优点就是性能，其他基本全是缺点。。

汇编语言，貌似也基本淘汰，比起机器语言，提升了很大的可读性。。

native语言(c c++) 相对汇编语言可读性又大幅提升，性能方面不如汇编，但比vm语言要高。。。缺点是开发效率，以及可移植性仍然需要提升。。很多图像库貌似都是使用此开发。。依赖于性能的应用大有所为。

vm语言(java  c#)  改进了native语言的问题，增加一个vm层隔离开了os。目前的业界高层应用开发主力

脚本语言 带来更高的生产力。但目前ide的问题，大型复杂企业级开发还难当重任，目前貌似在轻复杂度代码领域比较大发展。

Dsl语言 ，领域特定语言。。比如H5做界面。。Sql做数据库查询。Matlab halcon使用的语言，用来做图像处理。。

更高的开发效率，特别对于特定领域，图像处理，界面，数据库查询等大有优势。。正则表达式，用来文本搜索等。

Dsl语言本身性能很差，但是它是调用类库的，类库使用底层语言书写的，所以对性能不影响。

## **1.4.****混合编程 通过在高层语言嵌入dsl**

混合编程也是一个趋势，比如嵌入正则表达式文本搜索。嵌入sql数据查询。。使用h5做gui界面。。

图像处理领域，可以嵌入matlab halcon的dsl来进行图像处理。。游戏领域貌似当年编译语言嵌入lua脚本很流行，现在貌似嵌入js了开始。。

从目前趋势来看，嵌入dsl是个更好的选择。术业有专攻嘛。。综合起来就是王道。

## 业务逻辑的dsl 目前还没流行，有待于未来的发展

工作流可以充当业务领域的dsl，但目前还较为繁琐。

## 目前语言排行榜

# Gui界面开发

尽可能的一般使用客户端h5技术了，目前趋势了，最通用的跨平台ui dsl。通用于桌面gui，web架构，以及安卓ios 跨平台与hybrid模式app

html工具使用dw等可视化拖拉式设计界面比较常用，js方面就webstorm等这些了

界面直接html，中间通过服务端数据获取，ajax，注入浏览器对象等模式来与后端沟通.获取到数据后，绑定到form界面或者表格控件等常见场景

最大限度的分离前后端。同时增强复用性。。

Html文件可以直接应用与java php net nodejs项目以及其他语言项目，以及安卓 ios等hybrid模式与跨平台app的开发。。

常用的行业框架有bootstrap jquery vue amazeui Framework7等。Anrular.js有点复杂，慎用

## Gui的发展历程与未来趋势

像素级》dom native 》》xml （安卓，wpf，javafx） 》》h5（dom）+ h5 canvas（像素级）

目前趋势来看，h5已经到了dsl级别了，到了一个比较高的层次了，未来发展暂时

## 客户端mvc 优先于服务端mvc

客户端的mvc更加容易调整，而无需服务端环境，有利于提升开发进度。

其次，页面流程调整也是要js来实现，方便调整。修改后只需要刷新即可，最大化方便开发。。

而且有着最大的复用性。。如果使用服务端mvc则难以打包为手机本地app或者其他语言项目内的重用等。

Vue.js angular.js等。。 Vue更加简单快捷。

## 客户端ui 优先于服务端ui

只是普通的表格table和表单form绑定数据推荐还是使用table+vue等绑定。

其他ui方面可以推荐amazeui等。

服务端的jfnail，jstl el jsp php等就不推荐了。

## 适当的全栈

如有时间，后端可以适当的了解下前端h5部分，普通的小的调整可以就自己完成，一方面完善自己的技术体系，一方面做起来项目更加轻松。

前端部分也是，可了解下后端，当然由于后端的复杂性，可以使用nodejs了解更加方便，都是js，一路通到底。。有时候需要服务器环境才可能方便前端调试，apache可以搭建一个最简单化的web服务环境。如需一些动态调整，Nodejs做后端服务器测试也方便。

# archi架构

## 常见架构发展历程（按时间 **主进化路线Cs》》 bs 》3层架构》 [SOA》rest》MSA(微服务架构](https://link.zhihu.com/?target=http://www.baidu.com/link?url=A2BxkXZ752i3XmxS3RdircaM30kFmcQGnZHap5ZgiRQWufdnuGq_cj57ATg-l7lg9EMFRHDMsMuusBL6KWLgM_" \t "https://www.zhihu.com/question/20657269/answer/_blank)**

## 项目结构 为了方便，尽可能优先使用模块》service/dao结构

主要是使用模块包作为优先级，方便跨项目的模块复用。在同步工具里面很容易的同步过去。

比如原来是com.xxx.service com.xxx.dao 这样的包

可以变成 com.xxx.user.service com.xxx.user.dao ,或者因为这样分模块以后，里面的类数量很少了，也可以直接放入 com.xxx.user包即可。。

前端模块也尽可能采用模块自包含结构，方便重用。一个模块文件夹包含自身的 html,css,js小图片等。框架类库模块等放入外部集中模块即可

## Hybrid混合框架模式

常常我们在已有项目中新增功能，或者模块。。此时新的功能就可以使用更为轻量级快速的技术。。原有模块使用的框架尽可能先不动。

## 中间端ioc的实现spring guice ServiceLoctor  ，解决spring启动慢的问题

spring 主体ioc就是一个bean工厂模式+xml可配置化

一般使用spring即可。不过可能spring稍微繁琐下。其次spring启动性能可能比较慢，大的项目可能一俩分钟有可能。

项目环境下，也很少需要更换组件模式的，所以可配置化的意义大打折扣，所以最新的spring已经支持注解模式，注解其实就是硬编码了。。产品模式下，则可配置化比较重要

如果需要更快的效率，可以使用google guice也是轻量级的ioc框架

或使用静态工厂模式，更快的开发，有ide的支持提示。也启动性能更快，几秒启动。

或者，实现ioc还有 ServiceLoctor 模式。。

Ioc也是可以混合多个ioc模式与框架来实现功能的。尽可能的优先使用轻量级的ioc。。如旧的模块使用spring，新模块可以使用guice ServiceLoctor 模式等都是可以的。

## 架构方面的提升 适当的简化架构

双层，三层模型混合使用。。对于难度大的模块使用复杂架构。。大量简单模块直接使用双层模型。。

# 数据存储

## 数据存储orm方面 apache dbutil》mybatis》hibernate

数据库有mybatis hibernate 这类的orm框架

apache dbutil最简便， 然后Mybatis 。。Hibernate相对比较重和复杂些。如有可能，尽可能优先于使用简单的框架

如在现有项目新增功能或模块，则可以使用混合多个orm框架。。比如原项目是hibernate的，新模块可以使用mybatis实现，这样可以加快新模块的效率

## 适当使用sp存储过程，触发器，视图。。大力简化开发

大力简化开发，以及提升可读性，或者性能。

## Sql hql vs 程序代码

Sql属于dsl 4gl ，程序代码属于3gl 。。

从开发效率的进度，优先使用sql。。Hql就是私有化的sql dsl了

至于3gl，就较为繁琐了。。

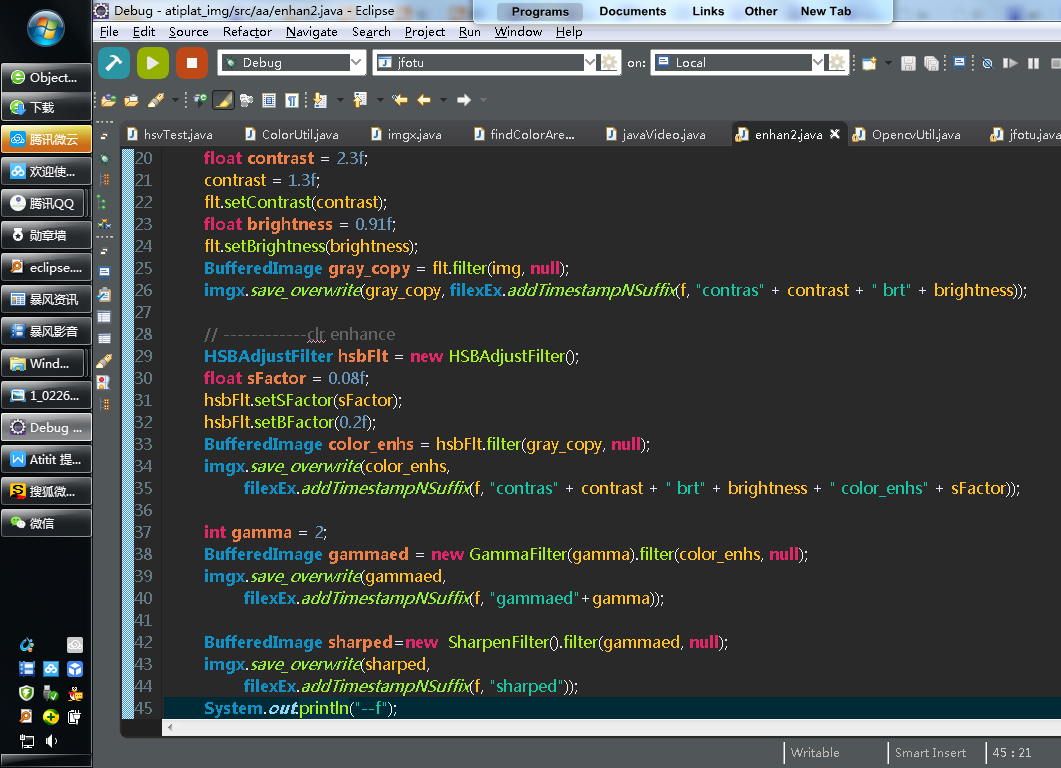
# dev开发

## 热部署对提升开发效果很有用 tomcat+ jrebel，或者resin。。

修改了代码后，不用重启web服务器即可生效。大大提升效率值得推荐

常常大项目需要一俩分钟启动可能。。

## 任务栏拉到左边，可以给宝贵的上下空间挪出一点地方



## Ide黑色背景主题不错

## 通用的轻量级编辑器，有语法提示notepad++ ，editplus Sublime 等

编辑配置文件，一些轻量级编辑场景下很合适。。

Sublime + h5插件貌似已经算是中级编辑器了

重量级ide那就dw webstorm eclipse等了。

## 其他，还有待于搜集

# 硬件

## 大屏幕 ssd 多核cpu

屏幕俩种，一种正屏 一种宽屏

正屏可以看到的上下空间更多一些 对提升工作效率很有帮助。

## 参考资料

Atitit. Gui控件and面板----程序快速启动区--最佳实践Launchy ObjectDock-o0g - attilax的专栏 - 博客频道 - CSDN.NET.html