

# 浅谈插件化软件开发

李鸣亚<sup>1</sup>, 邹晓峰<sup>2</sup>

(1.上海交通大学机械与动力工程学院, 上海 200030; 2.江南大学机械工程学院, 江苏 无锡 214036)

【摘要】论述了插件化软件开发的实现过程。通过对插件化软件开发概念的介绍, 总结讨论了插件化软件开发的优点, 并对插件化软件开发的原理进行了分析, 最后总结描述了插件化软件开发的现状。

【关键词】插件 软件开发 OSGI JPF

【中图分类号】TP311

【文献标识码】A

【文章编号】1003-2673(2009)08-81-02

## 1 引言

计算机硬件系统中有各种各样的可插拔的组件, 比如 CPU、硬盘、内存、……, 这些组件一起组成了完整的计算机系统。当需要升级时, 我们可以方便的更换其中的一个或几个, 只要新的组件符合主板的规范, 就可以正常工作。那么, 在软件系统中是否也可以像硬件系统那样实现“即插即用”? 插件化软件开发就是这种想法的产物。

## 2 插件化软件开发的概述

### 2.1 什么是插件?

软件系统中的插件 (Plug-in) 借鉴了硬件系统中插件的思想。它是一种符合一定规范, 实现了特定接口的程序, 主要是用来扩展软件的功能。当然, 插件的运行是需要一定的软件环境的。

插件技术并不是新的技术, 早期很多基于 COM 的开发的 Win32 应用程序都是插件化的系统, IE 浏览器就是一个典型。而 IE 浏览器下各种工具栏, 如 google 或 yahoo 等搜索栏就是插件了。

### 2.2 什么是插件化软件开发?

所谓插件化软件开发就是将软件系统分成一个主系统和若干个子系统, 主系统作为一个容器, 为插件提供运行的环境, 而将这些子系统做成相对独立的插件。

## 3 插件化软件开发的优点

### 3.1 可维护性好

当一个系统功能需要升级或者发现 bug, 我们可以做到修改尽可能小的一部分代码。这样一来我们会有两方面的好处, 一是节省了维护的成本, 二是修改后的代码给整个系统带来的风险最小。

### 3.2 可扩展性好

任何软件产品都是有生命周期的, 作为程序的创造者, 大家都不愿意看到其消亡的现实。因此应当尽量延长软件产品的生命周期, 不断的将新的功能注入到现有的软件产品是一种手段。插件化的系统架构为扩展新功能提供了更好的方便性。因为新的功能以插件的形式给出时, 是相对独立的, 不影响整个系统框架和其它功能。甚至是把扩展软件功能的能力提供给

用户或第三方厂商, 以达到最大限度的功能扩展。

### 3.3 适合团队开发

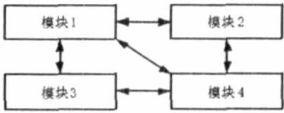


图1 普通模块划分

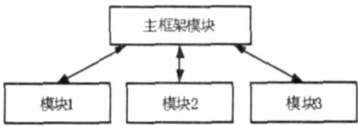


图2 插件化模块划分

图1是普通开发模式中的模块划分示意图, 我们可以看出各个模块之间的耦合度比较高。相比之下, 图2中插件化模式下的模块划分, 因为各个子模块只与主框架模块发生联系, 模块之间的耦合度低很多。在团队开发中, 如果将各个模块分配给不同的成员开发, 那么图2在开发中的复杂度势必低许多, 相互之间用来沟通所用的时间成本也低。

## 4 插件化软件开发的原理

插件化软件系统可以划分为主框架+插件两个部分。主框架是系统的核心和基础, 它的主要功能应为内核功能和插件管理功能。插件所完成的功能是对主框架功能的扩展与补充。在确定用插件化方式开发一个系统后, 需分析哪些功能应该由主框架实现, 而哪些功能应该由插件实现。

为实现主框架+插件结构的软件设计需要定义两个标准接口, 一个是主框架实现的主框架扩展接口, 一个是插件实现的插件接口。主框架扩展接口完全由主框架实现, 插件只是调用和使用, 插件接口完全由插件实现, 主框架也只是调用和使用。主框架扩展接口实现插件向主框架方向的单向通信, 插件通过主框架扩展接口可获取主框架的各种资源和数据, 包括各种系统句柄, 程序内部数据以及内存分配等。插件接口为主框架向插件方向的单向通信, 主框架通过插件接口调用插件所实现的功能, 读取插件处理数据等。

主框架插件处理功能包括插件注册、管理和调用。插件注册为按照某种机制首先在系统中搜索已安装插件, 然后将搜索

【作者简介】李鸣亚(1979-), 江苏无锡人, 上海交通大学机械与动力工程学院在读研究生, 无锡科技职业学院讲师。

到的插件注册到主框架上，并在主框架上生成相应的调用机制。插件管理完成插件与主框架的协调，为各插件在主框架上生成管理信息以及进行插件的状态跟踪。插件调用为调用各插件所实现的功能。

## 5 插件化软件开发的现状

目前，在 Java 中插件化开发主要有两个技术：OSGI 和 JPF。以下将简要介绍一下这两种技术。

### (1) OSGI

OSGI 是服务平台的规范。通过实现能够提供服务的符合 OSGI 规范的组件，用户可以将其组件发布到 OSGI 运行平台，供用户和其他组件使用。OSGI 组件提供的服务具有两个层面的含义：系统层面，即一个组件为其他组件提供服务，这些服务体现为 Java 接口的实现；业务层面，即一个组件为外部系统或用户提供某种业务服务的实现。在核心，OSGI 是一个组件和服务模型，如图 3 所示。

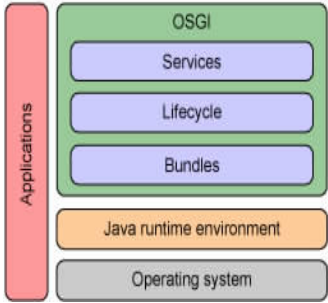


图 3 主机操作系统、JAVA 和 OSGI 中层的交互

(上接第 70 页)

失效。液压泵自身工作时振动很大，液压泵轴套及密封圈等密封件承受反复的微小变形，如果振动程度太大或密封圈使用一段时间后产生磨损，不能补偿振动引起的位置偏移，就会引起泄漏。

## 4 主要防治对策

造成液压泵泄漏的因素是多方面综合影响的结果，以现有的技术和材料，要想从根本上消除泄漏是很难做到的。只有从影响泄漏因素出发，采取合理的措施尽量减少泄漏。

(1) 合理的设计和加工。合理的设计是根本，设计时要考虑回油槽、回油孔、挡板等的位置、尺寸、及精度等；设计时还要选择适当的密封结构以及材质、功能与液压泵实际工作要求相符的密封件，加工过程中应注意密封面的最后精加工，必须严格遵守密封沟槽等的尺寸精度、相对滑动的外径精度、表面粗糙度等标准。

(2) 正确的装配。借鉴以往的经验，必须确认被密封的流体和密封件的关系，正确选择密封装置。装配中应注意密封件的安装方向，不能反装，必要时可涂抹润滑剂，增加润滑，以便安装，安装过程中应保持清洁，避免毛丝、切屑、灰尘等杂质进入密封处，安装时尽量使用专业工具，使用的工具不要有尖端、刃口，以免在安装过程中对密封件或密封面造成划痕等。

在 OSGI 的成功应用中，最有名的应该是 Eclipse。Eclipse 从 3.0 开始引入 OSGI 实现插件的支持。Eclipse 支持插件，人们可以根据它的规范开发自己需要的插件，以实现功能的扩展。

### (2) JPF

Java Plug-in Framework (JPF) 提供了一个动态发现和加载插件的实时引擎。这个框架用来开发维护成本低可升级的，可扩展的 Java 应用程序。JPF 可以嵌入到任何类型的 Java 项目中。

JPF 有不少成功的应用，其中比较成功的应用是 Logical-Doc，它是一个文档管理器。

## 6 结束语

插件化软件开发有很多优点，并且已经有一些成功的应用。但是，任何一个技术都不是完美的，它也存在一些不足，需要不断的完善。目前主要的成功应用几乎都是桌面程序，基于 web 的应用比较少。相信今后这项技术会不断的完善，具有良好的应用前景。

(3) 及时的维护和保养。液压泵应合理的使用，减少不必要的启动、停止的频率，控制液压油的污染，要从污染的源头入手，加强污染源的控制，还要采取有效的过滤措施和定期的油液质量检查，为有效地切断外界因素（水、尘埃、颗粒等）对液压油的污染，可加一些防护措施等；充分考虑液压泵工作条件，对液压泵的元件应制定检测周期与部位，如负载变形量、泵的振动量，以便控制变形量与振动量，对液压泵规定检测周期，特别是对密封件，应定时进行检测，及时发现并及时采取处理，才能保证液压泵的正常工作的。

## 5 结束语

液压泵中的泄漏防治必须找出引起泄漏的根本原因，全面考虑影响泄漏问题的使用条件、引起泄漏变化的因素等，才能真正做到行之有效地防治。

### 参考文献

[1] 黄迷梅. 液压气动密封与泄漏防治[M]. 北京: 机械工业出版社, 2003.  
[2] 蔡廷文, 张冰蔚. 机电系统故障分析与维护[M]. 北京: 化学工业出版社, 2005.  
[3] 赵应璩. 名优机械液压系统及其修理[M]. 上海: 上海交通大学出版社, 2002.