

文章编号: 0253-2697(2014)02-402-05 DOI:10.7623/syxb201402025

组件开发技术在大型测井软件平台研制中的应用

王才志¹ 李 宁^{1,2} 刘英明¹

(1. 中国石油勘探开发研究院 北京 100083; 2. 长江大学 湖北武汉 430100)

摘要: 组件式开发是中国石油新一代测井软件 CIFLog 研制中所运用的一项关键技术,可极大提高平台扩展的能力和灵活性,在油田属地化应用模块集成和二次开发中发挥了重要作用。根据组件概念和设计思想,制定了测井专用组件开发规范,重点介绍了组件的设计和实现方法,包括注册基类的设计、MVC 模式组件结构设计方法和组件事件分发机制实现方法。最后给出了利用 Lookup 机制对组件进行注册和调用组件的方法和过程。了解和掌握这一技术,对指导当前正在大规模开展的 CIFLog 现场推广应用具有重要意义。

关键词: CIFLog 平台;测井;组件;软件开发;MVC

中图分类号: P631.83; TP319 **文献标识码:** A

Application of component development technology in large-scale well logging software platform-CIFLog

Wang Caizhi¹ Li Ning^{1,2} Liu Yingming¹

(1. PetroChina Research Institute of Petroleum Exploration and Development, Beijing 100083, China;

2. Yangtze University, Wuhan 430100, China)

Abstract: Component development, which plays an important role in integration of oilfield localization application module and secondary development of oilfields, is a key technology to increase the extensibility and flexibility in the new well logging software platform of PetroChina-CIFLog. The development specification of special logging component is formulated based on the concept and design thinking of component. This paper mainly introduces component design and implementation methods, including the registration base class and model-view-controller (MVC) component structure design, and event dispatching mechanism. Finally, the methods and procedures of registering and calling components based on Lookup mechanism are presented. It is essential to understand and master this technology, which has great significance in guiding the extensive promotion and application of CIFLog.

Key words: CIFLog platform; well logging; component; software development; MVC

CIFLog 测井处理解释软件平台采用跨平台 Java 编程语言和 Netbeans 框架进行开发,实现了数据管理与资料应用、单井解释与多井评价、裸眼井解释与套管井监测一体化,以满足复杂储层测井综合评价的需要^[1]。在平台的开发过程中已经形成了一系列关键技术^[2-4],笔者所介绍的测井专用组件开发是 CIFLog 平台开发中的又一项关键技术。

对于系统复杂、结构庞大的软件平台,软件的健壮性、开放性、可扩展性和易维护性是必须考虑和解决的关键问题,提出组件式开发思想是为了使计算机软件开发更符合人类的行为方式,通过开发各种各样功能专一的组件,将它们按照功能要求组合起来,进而形成复杂的测井处理解释应用系统,提高开发效率^[5-9]。不

同的编程语言和应用范围有不同的组件开发技术,例如 Windows 操作系统平台上的 COM 组件开发技术,Java 语言的 JavaBean、JavaApplet 组件开发技术等。笔者针对测井专业处理解释软件特点,采用 Java/J2EE 设计模式,形成了测井专用组件开发技术。该技术为平台快速开发提供了技术保障,也有助于平台开发过程中编程人员的并行工作,是平台开发的核心和关键技术之一。

1 设计规范与组件结构

测井处理解释软件平台由基础工具模块、测井资料预处理工具模块和各种处理解释应用模块组成,这些模块中许多功能相同或近似,例如数据源选择、曲线

基金项目: 国家重大科技专项(2008ZX05020)资助。

第一作者及通信作者: 王才志,男,1970年10月生,1992年获西安石油学院矿场地球物理专业学士学位,2004年获中国石油勘探开发研究院矿产普查与勘探专业博士学位,现为中国石油勘探开发研究院高级工程师,主要从事测井处理解释方法研究与软件开发。Email: wangcaizhi@petrochina.com.cn

选择、参数编辑、表格显示、曲线显示等等,这些相对独立的功能开发完成后,如果能够重复使用,无疑会大幅提高开发效率。而在程序开发中,组件技术正好可以满足这样的要求。组件的含义就是对数据和方法进行简单封装,并使其成为具有特定功能的对象,在以后开发过程中重复使用。

组件在设计时要注重宏观层次上的划分,保证组件之间划分的合理性和依赖性。组件的设计没有通用的准则,稳定性和可重用性是组件开发中首先考虑的问题之一。组件在具体设计开发过程时,还需要有明确的开发规范,保证组件的灵活性和可重用性。在 CIFLog 平台组件开发过程中,根据测井软件平台组件的特点和使用方式,制定了 CIFLog 平台的组件开发规范:①组件开发结构类型是父子关系结构,一个组件可以包含任意层次的子组件;②基于统一的父接口,实现组件在 Lookup 组件管理器中的注册;③通过公共功能接口,实现对组件的控制;④组件开发要符合模型—视图—控制器 (Model-View-Controller) 设计模式,以保证外部提供的数据模型在组件中可以使用;⑤组件提供事件分发功能,实现与应用程序之间有效的工作协同。

按照以上开发规范,设计了组件的框架和结构,如图 1 所示。CIFLog 平台中均采用该框架结构进行组件设计和实现。

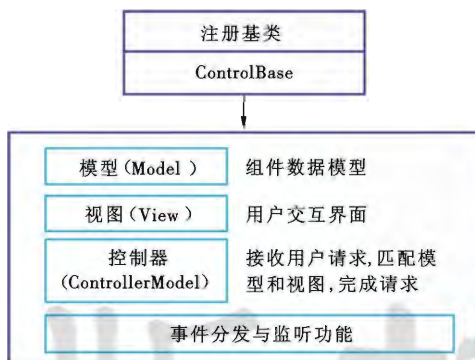


图 1 组件设计结构框图

Fig. 1 Structure diagram of component

2 组件开发

组件开发过程中,要根据实际应用和不同粒度组合需要规划组件的功能,按照组件的结构框架(图 1)对组件进行规范开发,以保证可重复性。

CIFLog 平台众多组件中,测井绘图组件包庞大和复杂,在绘图组件包中包括:绘图面板、画布、工具栏、右键菜单、线性道、对数道、一维曲线、填充曲线、成像曲线、波形曲线等几十种组件,而其中曲线组件最具

代表性,下面将以曲线组件为例进行介绍。

2.1 设计注册虚基类

一般来说,一组功能相近的组件需要设计一个注册虚基类,组件封装类必须继承自该注册虚基类,这样就使得组件有了一组规范的接口。

在测井绘图中,一维曲线、填充曲线、成像曲线、波形曲线等是一类组件,功能相近,需要为曲线组件设计统一的注册虚基类,在类中提供用于注册的基本信息接口和创建函数接口。在设计注册虚基类时,关键是抽提出一组规范接口,这样基于注册虚基类开发出的组件在后续应用程序开发中才可以进行注册和调用。

下面给出了曲线绘制注册虚基类的部分代码。

```
public abstract class CurveRegisterBase {
    public abstract ImageIcon getNodeIcon(); // 获取图标
    public abstract String getTipName(); // 获取提示信息
    public abstract int getCurveType(); // 获取曲线类型
    public abstract String getIdentityName(); // 获取唯一性标识
    public abstract BaseShape createShape(); // 创建绘图对象
    public abstract String getSimpleName(); // 获取短名称
    public abstract String getAliasName(); // 曲线别名
    public abstract boolean displayOnMenu(); // 是否在菜单中显示
    public abstract String getGroupName(); // 曲线分组名称
    public abstract ImageIcon getGroupIcon(); // 曲线分组类别图标
    .....
}
```

在测井绘图中,因为测井曲线类采用了相同的注册虚基类,采用文章后面介绍的 Lookup 方法,应用程序就可以实现曲线注册和调用,并且在不修改原有代码情况下,扩充测井曲线绘制类型。平台正是基于这样的实现方式,在裸眼井测井图件绘制的基础上,分别扩充形成了生产测井图件绘制和水淹层测井图件绘制系统。

2.2 MVC 设计模式

应用 MVC 设计模式,有助于提高软件的可扩展性和代码可重用性^[10-11]。在测井绘图中,同一种曲线

显示方式、数据来源可能不同。例如图2中所示的棒状图,数据可以来自常规一维测井曲线数据(A),也可以来自离散实验分析数据(B)。要想简化程序设计,必须减少数据表达、数据描述和应用操作的耦合度,采用MVC模式屏蔽数据不同来源,重用绘制代码。

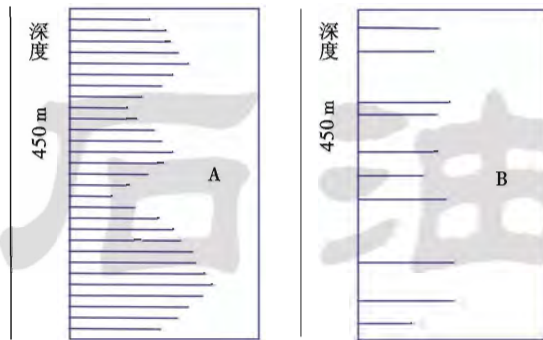


图2 一维测井曲线和离散数据棒状图显示

Fig. 2 Bar graphs of one dimensional curve and discrete data

MVC 全名是 Model View Controller,是模型—视图—控制器的缩写。图3给出了模型、视图、控制器三者之间的关系。其中,模型是组件的主体部分,表示业务数据,或者业务逻辑;视图是用户看到并与之交互的界面;控制器可根据用户的输入,控制用户界面数据显示和更新 model 对象状态。

在测井曲线组件开发中,模型的设计非常重要。通常先建立一个默认的模式 DefaultModel, Default-Model 基本满足测井曲线组件设计功能要求。随着组件应用的扩展,其他地方也要用到该组件,此时组件使用的数据来源和业务逻辑发生了变化,例如数据可能

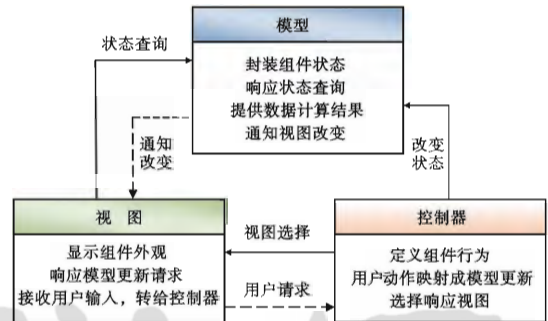


图3 模型、视图、控制器之间的关系

Fig. 3 Relationship of model-view-controller

来自其他类型测井曲线、数据表、岩心分析资料等,这时需要重新设计一个模型(NewModel),在 NewModel 中实现新的业务逻辑和数据访问。应用时,首先创建 NewModel 类的实例对象,将该对象通过接口函数设置给组件,然后,替换掉原有的 DefaultModel。由于 MVC 的3个部件相互独立,改变 Model 不会影响其他两个,所以依据这种设计思想能构造良好的松耦合构件,达到扩展组件功能,实现组件重用的目的。

由于采用了 MVC 设计模式,测井绘图仅负责显示和交互操作,数据从什么地方来已经变得不重要,使得测井绘图应用范围大大提高,只要对接好数据,也就是实现新的模型,就可以直接显示其他文件数据或来自远程数据库的数据,更好地体现了组件的作用和价值。

2.3 事件分发机制

曲线组件鼠标交互操作后,需要通知其他组件响应该操作,也就是组件需要具备事件分发和事件处理的能力。图4显示了处理参数层段列表和深度曲线之

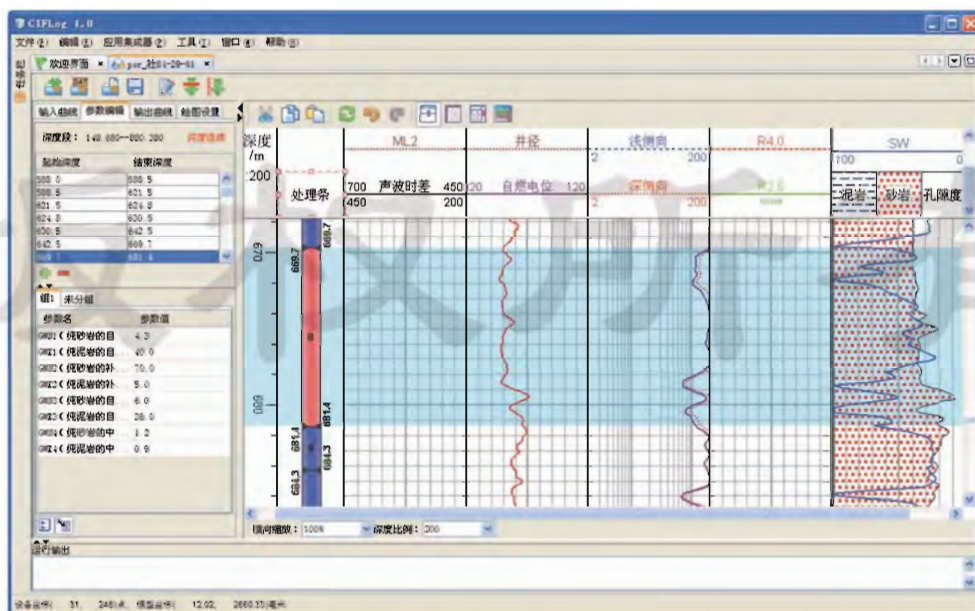


图4 处理参数层段列表与深度曲线交互操作的响应与协同

Fig. 4 Interaction response and collaboration between processing parameter interval list and depth curve

间交互操作的响应与协同。当在测井曲线图上用鼠标改变一个深度层段,在处理参数列表中要立即更新该深度值,而当在处理参数列表中修改一个深度值时,在测井绘图的深度曲线图上也要更新显示。在以往开发中,两部分要相互调用,必须互相依赖,使得程序耦合度增加,开发困难,而通过分发事件方式,可以很好解决响应和协同问题^[12-13]。

在组件中,用户在点击按钮或执行特定命令时,将向应用程序发送相关的消息,应用程序通过事件监听器对象,捕获组件分发的消息,并对此做出响应。事件分发功能的实现包括以下几个方面:

(1) 事件,用一个事件对象来表示该操作或动作命令,在上例中分别定义深度曲线层段修改事件和层段深度值改变事件。

(2) 事件源,产生事件的组件。在图 4 的应用程序中,分别调用深度曲线和层段列表注册方法把应用程序作为事件监听者注册给深度曲线和层段列表,让深度曲线和层段列表成为事件源,当事件源发生事件时,事件监听者(应用程序)就代替事件源对事件进行处理,实现委托型事件处理模式。

(3) 监听器,当事件发生,应用程序作为事件监听者处理深度曲线和层段列表的事件,由应用程序完成响应与协调,组件之间没有任何关联。

3 组件调用方法

经过长时间摸索,发现 NetBeans IDE 中的 Lookup API 访问可提供扩展服务,使得不同组件的注册可

以轻易进行,在平台各个模块中查找组件也较为容易,是一套简单而适用的解决方案。

Netbeans 提供了一个默认 Lookup,它一般用于对特定位置指定的类进行注册服务,注册信息存放在 Java 运行包 jar 文件的 META-INF/services 目录中。Lookup 实际上就是一个 Map 类,其中的键就是类对象,而值就是扩展获实现键一类的一组类对象^[14]。该方法的一大优点就是可以对提供某些对象的组件和使用这些对象的组件之间进行解耦,可以在不同的模块中实现,而系统的其余部分将继续透明地工作。图 5 给出了组件注册和调用示意图。

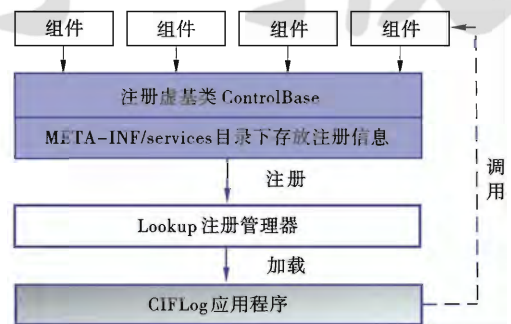


图 5 组件的注册和调用

Fig. 5 Registration and invoking of components

组件注册和调用的方法如下:①在 Netbeans IDE (集成开发环境)中,在组件所在的模块下创建 META-INF/services 目录,在该目录下创建文件名为基类类名称的文件;②在创建的文件中写入组件类名称;③平台运行时 Lookup 监控 classpath 类路径,检查到

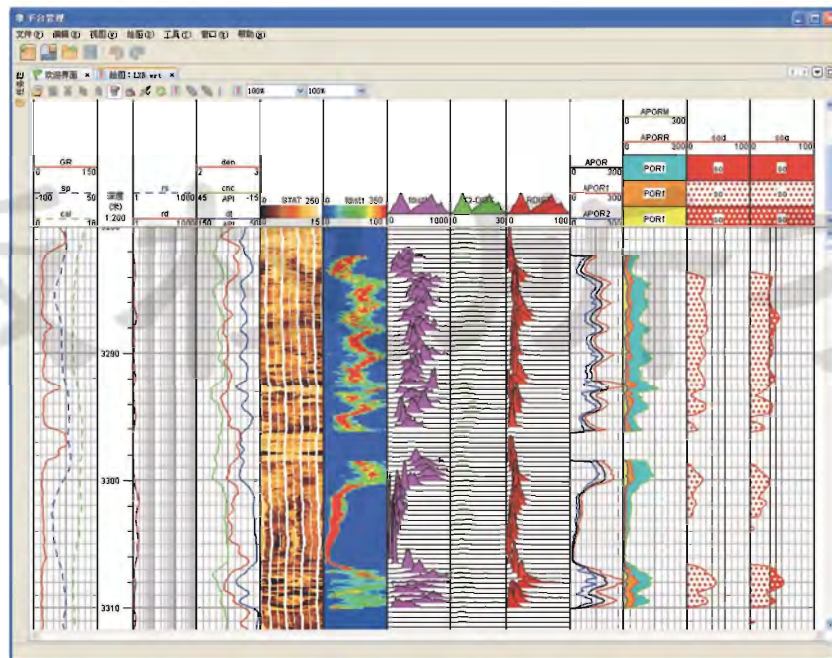


图 6 测井绘图中曲线组件的显示

Fig. 6 Display of curve components in drawing module

所有注册的组件;④通过调用 Lookup 对象的 lookupResult 方法,获得一个包含有多个组件的 Result 实例,再调用 Result 实例的 allInstances 方法便可获得组件的整个集合。通过迭代就可以获得各个组件对象进行使用。

由于 META-INF/services 下的注册信息在生成 Java 运行库 Jar 时,会打包到 Jar 中。所以只要按照 CIFLog 规范开发的组件,在生成 Jar 时包含了 META-INF/services 注册信息,就可以动态地加载到 CIFLog 平台中,实现平台组件扩展。

利用组件开发技术,平台实现了各种曲线组件的开发,图 6 是曲线组件在测井绘图中的显示。若现有的曲线显示满足不了用户需求,利用以上介绍的技术就可以设计新的曲线组件,扩展已有的可绘制曲线类型。

目前,根据测井处理解释软件平台的特点,平台规划、设计并实现了如表 1 中所列的众多组件,平台中所有模块都是基于这些组件开发完成的。

表 1 CIFLog 平台中的组件

Table 1 Components on CIFLog platform

数据源选择对话框	线属性设置组件	曲线头切分
文本编辑器	字体属性设置组件	图元撤销恢复
自定义文本字段组件	填充属性设置组件	绘图导航条
参数表格组件	颜色选择框	注释图元选择组件
进度条	色标选择控件	图元剪切/复制/粘贴
深度控制条	自定义按钮	曲线数据点拾取组件
特殊字符选择面板	绘图属性面板	曲线选择工具
带记忆功能的文件选择器	绘图数据源对话框	曲线深度显示比例
宏管理器	左右刻度属性设置组件	切换组件
模板\卡片文件选择对话框	无效值过滤组件	绘图画布缩放控制组件
参数卡解析器	线性渐变填充画布	绘图图元导出图片组件
表达式解析器	渐变组件面板	图元对齐方式选择组件
XML 解析器	岩性符号选择组件	绘图标尺组件
文本路径解析器	解释结论选择组件	

4 结 语

基于测井专用组件开发规范,利用 MVC 模式对组件结构进行设计,提高了组件可重用性和易维护性,采用事件分发机制提高了组件交互能力。基于以上技术在 CIFLog 平台中开发的组件,通过组合可以快速形成新的应用模块,并且可以根据油田不同的应用习惯进行属地化改造,为平台系统升级和属地化应用系统开发提供了有力的技术支持和保障。

参 考 文 献

- [1] 李宁,王才志,刘英明,等. 基于 Java-Netbeans 的第三代测井软件 CIFLog[J]. 石油学报,2013,34(1):192-200.
Li Ning, Wang Caizhi, Liu Yingming, et al. CIFLog: the 3rd generation logging software based on Java-Netbeans[J]. Acta Petrolei Sinica, 2013, 34(1): 192-200.
- [2] 赵亦朋,王才志,黄亭宇. 基于 SOA 的测井解释平台中间层设计

[J]. 石油学报,2008,29(5):782-785.

Zhao Yipeng, Wang Caizhi, Huang Tingyu. Design of SOA-based middle ware for logging interpretation platform[J]. Acta Petrolei Sinica, 2008, 29(5): 782-785.

- [3] 夏守姬,李宁,李伟忠,等. Java 测井平台上多语言应用集成框架设计[J]. 石油学报,2010,31(5):810-814.

Xia Shouji, Li Ning, Li Weizhong, et al. Design of integration framework for multi-language applications on logging platform in Java[J]. Acta Petrolei Sinica, 2010, 31(5): 810-814.

- [4] 刘英明,李宁,张玲,等. 基于 JNI 技术 C++ 测井应用程序集成方法[J]. 石油学报,2010,31(6):980-984.

Liu Yingming, Li Ning, Zhang Ling, et al. Integration of the C++ well logging application based on JNI technology[J]. Acta Petrolei Sinica, 2010, 31(6): 980-984.

- [5] 戴敏,宋燕平. 组件开发与软件重用[J]. 南开大学学报:自然科学版,2001,34(3):24-27.

Dai Min, Song Yanping. Component developing and software reuse[J]. Journal of Nankai University, 2001, 34(3): 24-27.

- [6] 荆心,张晓棠. 可复用软件组件开发框架研究[J]. 西安工业大学学报,2007,27(5):464-467.

Jing Xin, Zhang Xiaotang. Study on the frame for reusable software component development[J]. Journal of Xi'an Technological University, 2007, 27(5): 464-467.

- [7] 刘治国,赵平,张玲,等. COM 技术在测井软件开发中的应用[J]. 测井技术,2004,28(3):249-250.

Liu Zhiguo, Zhao Ping, Zhang Ling, et al. Application of COM in log system software development[J]. Well Logging Technology, 2004, 28(3): 249-250.

- [8] 李洪奇. 基于知识的组件化测井软件开发框架[J]. 计算机工程与应用,2006,42(12):229-232.

Li Hongqi. Designing a logging domain framework with component model base on knowledge[J]. Computer Engineering and Applications, 2006, 42(12): 229-232.

- [9] 李洪奇. 测井软件构件化开发框架[J]. 测井技术,2005,29(5):399-403.

Li Hongqi. Architecture development of well logging software based on components[J]. Well Logging Technology, 2005, 29(5): 399-403.

- [10] 田新丽,白绳武,董秋鸿. 基于 MVC 模式的地图控件的设计[J]. 电子科技,2010,23(6):41-42.

Tian Xinli, Bai Shengwu, Dong Qiuhong. Design of map control based on the MVC pattern[J]. Electronic Science and Technology, 2010, 23(6): 41-42.

- [11] 王全彬. MVC 架构模式在 Java 开发中的应用[J]. 四川理工学院学报:自然科学版,2009,22(1):38-40.

Wang Quanbin. Application of MVC framework in the development of Java[J]. Journal of Sichuan University of Science & Engineering: Natural Science Edition, 2009, 22(1): 38-40.

- [12] Eckel B. Java 编程思想[M]. 第二版. 侯捷译. 北京:机械工业出版社,2002:519-525.

Eckel B. Thinking in Java [M] Second Edition. Translated by Hou Yi. Beijing: China Machine Press, 2002: 519-525.

- [13] Topley K. JFC 核心编程[M]. 第二版. 郭旭,朱洁斌,吴宇文译. 北京:清华大学出版社,2003:68-73.

Topley K. CORE JFC Java Foundation Classes[M]. Second Edition. Translated by Guo Xu, Zhu Jiebin, Wu Yuwen. Beijing: Tsinghua University Press, 2003: 68-73.

- [14] Boudreau T, Tulach J, Wielenga G. Rich client programming: plugging into the netbeans™ platform[M]. Upper Saddle River: Prentice Hall, 2007:49-68.

(收稿日期 2013-09-17 改回日期 2013-12-10 编辑 宋宁)