

一种开放式水力机械状态数据应用平台^①

周 叶, 潘罗平, 谭志锋

(中国水利水电科学研究院, 北京 100038)

摘 要: 本文针对水力机械状态监测数据的多样化应用需求, 设计了一种开放式模块化数据分析与应用平台, 并提出了这种软件平台的设计思路、功能组成与实现原理, 对模块标准化和模块间数据共享的难点, 采用动态链接库和内存共享方式给出解决方案. 通过开放模块的模板和接口, 能够实现多方协作开发集成, 减小系统功能的耦合, 最终为水力机械的数据挖掘提供易用的软件框架与基础.

关键词: 水力机械; 数据应用平台; 动态链接库; 内存映射

Open Condition Data Application Platform for Hydraulic Machinery

ZHOU Ye, PAN Luo-Ping, TAN Zhi-Feng

(China Institute of Water Resources and Hydropower Research, Beijing 100038, China)

Abstract: For the needs of various applications with condition monitoring data of hydraulic machinery, the paper describes an open modular data analysis and application platform, and provides the design, functions and application principles for the platform. Which are difficult to implement just like modules standardizing and data sharing with modules, it gives solutions with Dynamic Link Library and memory mapping. By using open form framework and API function, we can make it possible for collaborative development and decrement of system functions coupling. The platform can provide a basis and framework for apply different data mining technologies of hydraulic machinery.

Key words: hydraulic machinery; data application platform; dynamic link library; memory mapping

近年来水电机组状态监测与诊断技术得到了迅速的发展, 也在水电行业得到广泛的应用, 一定程度上为电站的安全高效生产与运行管理发挥了重要的作用. 但随着状态数据的采集和积累, 简单的趋势分析和实时监测已无法满足数据挖掘、故障诊断和状态预估的要求, 因此, 围绕着这些机组状态数据, 各种分析应用系统软件层出不穷, 而电厂工程师或应用人员仍会提出各种要求, 需要系统进行完善和改进^[1].

一方面是数据应用需求的多样化, 另一方面是状态监测与分析系统的市场化, 一种将常规的状态数据采集、存储和应用平台的标准化设想就提了出来, 如果以状态数据为核心, 整个数据应用的流程便简化为了数据采集、数据存取和应用平台三部分内容, 而前两部分, 完全可以通过数据存储格式标准、数据库存

储标准、数据读取标准三项标准的制定来规范和统一化. 而第三个环节—应用平台的设计和实现, 则是标准化的重要环节, 也是难点.

通过设计通用的水力机械状态数据应用平台, 开放功能模块的编制模板和接口函数, 既可以便于多方协作开发集成, 又减小了系统的耦合性, 避免了重复开发和建设, 使得应用开发更具有专业性和实用性.

1 平台框架设计

1.1 应用平台的组成

由于对水力机械的监测数据可以进行多种分析和处理手段, 数据来源也各不相同, 因此对开发者来说, 不可能提供用户想要的全部功能. 因此, 整个应用平台的设计必须按照模块化组合, 即按照一定的标准,

^① 收稿时间:2013-04-01;收到修改稿时间:2013-05-06

既可嵌入第三方应用程序,也可以通过提供标准接口,让电厂用户或专业开发人员进行定制和模块开发。

应用平台分为框架和模块两部分组成,模块由各种功能界面组成,如状态监测模块、趋势分析模块、报表生成模块等;而框架部分,则是这些模块的容器,用来提供读取各模块的描述、图标等信息,提供模块的归类 and 菜单显示,并为这些模块提供应用和展示的平台。

应用模块通过插入的方式集成到平台框架中,按照模块的开发集成标准,由第三方开发模块 DLL 文件,框架通过调入函数将对应的模块调入。通过组合不同的模块 DLL 文件,同一个平台框架程序,可以封装为不同的数据应用系统,如水电机组状态监测系统、水电机组趋势分析系统等,而某一系统中模块的使用,则由登录用户权限进行管理和限制。

在用户使用客户端软件时,一旦需要新的功能,只需由第三方按照标准开发新的功能模块,由用户插入至框架中即可,如果不需要某功能,只需直接将该 DLL 文件移除。

模块和框架都需要遵守统一的设计标准,而模块与框架间只存在结构的关系,功能模块在框架调用后,并不再与平台框架打交道,需要的数据直接通过接口函数进行存取,并独立执行模块的程序功能。

1.2 平台框架的功能

与分析模块不同,平台框架除了作为各模块的容器,自己仍然需要具备一定的框架通用功能,首先是模块的管理,平台框架已经预置了几个模块功能分类,分类名称可以配置。有些分析功能就归入同一个“数据分析”分类菜单下,而模块的设置功能,则置于“系统设置”的分类菜单下,这样通过模块的管理和分类,可以实现系统各种模块和功能的有序和易用性。

权限管理是平台框架的另一个重要功能,尤其是水力机械的状态数据很多,有电气的、机械的,使用系统的用户来源和职责也不同,对同一套系统,需要根据不同用户权限实现功能的区分,因此通过读取存取在数据库中的用户权限信息,来实现菜单上对应功能模块的隐藏和显示。

同一个平台框架通过不同模块 DLL 的组合,形成各种不同用途的系统,系统通过皮肤套件和配置文件的选择,实现不同系统采用不同的显示样式,这样可以同一套框架显示完全不同的色彩、样式和风格。

而资源链接、模块升级等功能,都是平台框架为了定制和管理所必须的功能,虽然这些不涉及到专业的分析和应用,但却是关系到平台框架易用性的基本功能。

2 平台框架实现

平台框架采用 Embarcadero 公司的 Delphi XE2 开发,开发语言采用 Object Pascal,由于平台框架仅仅作为功能模块的容器,故采用 MDI(Multiple Document Interface)形式。由于 MDI 形式可以在平台框架内,允许多个用户界面同时并存和运行,因此更适合水力机械数据分析中,不同分析结果的对比,以及状态监测时多监测界面的展示。

2.1 模块 DLL 的加载

由于平台框架与模块间采用 DLL 方式载入,因此,首先需要确定的是 DLL 的载入方式。当前在 Windows 系统编程中, DLL 文件的载入方式有两种,分别是静态调用和动态调用^[2,3]。

静态调用是访问外部例程最容易和最常用的方式,即通过名字或索引号进行 DLL 中函数和过程的加载。由于该种方式下,程序启动时会加载所有引用的 DLL,因此如果找不到对应的 DLL,程序将无法启动,在程序拥有大量的 DLL 模块文件时,给程序的启动和用户体验带来影响。

而动态调用是指在需要时才进行 DLL 文件的加载,并在使用完毕后,及时释放该 DLL 文件。因此平台框架的开发采用了动态调用的方式,该种方式需要用到 Windows 的 API 函数 LoadLibrary、GetProcAddress、FreeLibrary 等。

以调用 DLL 中的模块描述和菜单分类序号为例:

```
TDLLDescribe = procedure(var Desc: string; var FormTag: Integer); stdcall;
```

这里定义的 TDLLDescribe 其实就是 DLL 文件中要调用的某过程的指针。在动态调用 DLL 时,首先声明:

```
LibHandle: THandle;  
DescribeProc: TDLLDescribe;
```

也就是将加载的 Module.dll 文件作为句柄变量,并获取该句柄中对应函数的地址,赋给 DescribeProc,就可以通过使用 DescribeProc 起到调用 DLL 中函数的作用了。整个过程语句如下:

```
LibHandle := LoadLibrary('Module.dll');  
DescribeProc := GetProcAddress(LibHandle,
```

DescribeDLL);

DescribeProc(Description, FormTag);

FreeLibrary(LibHandle);

最终通过 FreeLibrary 函数释放该 DLL 文件,做到即时调用和释放内存资源.

2.2 动态菜单的生成

根据检索配置文件夹中的不同模块 DLL 文件,平台框架自动获取这些 DLL 的描述和图标,并生成动态菜单,当然菜单项的显示与否,也需要登录的用户级别来进行限制.

由于菜单项在检索多个 DLL 文件后生成,因此,在点击动态生成的菜单项时,需要调用对应的 DLL 文件中界面生成函数.也就是读取 DLL 信息时,首先调用 NewItem 函数,生成对应的菜单子项,语句如下:

```
aMenuItem :=NewItem(Description, 0, False, True, AddMenuClick, 0,MI_'+inttostr(i));
```

这里的 Description 来自动态加载 DLL 时 DescribeProc 函数中获取的该 DLL 的描述,而 AddMenuClick 则为该菜单子项的点击事件,在点击事件中,仍然是动态加载 DLL 的代码,不同的是,调用的是 DLL 中创建窗体的 CreateDLLForm 函数.

```
@DLLSub :=GetProcAddress(DLLHandle, 'CreateDLLForm');
```

这样整个平台框架中,从检索模块 DLL 到添加菜单项,全部以动态的方式实现,完全可以根据不同的配置文件夹,实现不同的应用系统切换.

整个系统动态加载 DLL 的流程如图 1 所示:

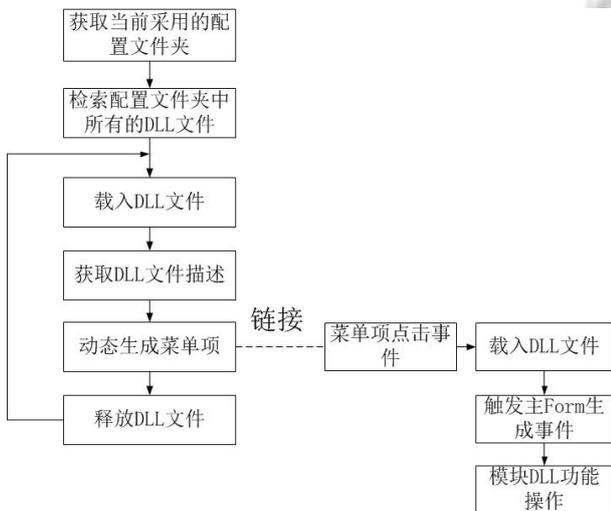


图 1 系统动态加载 DLL 流程

3 功能模块标准化

这里的功能模块主要指 DLL 文件(动态链接库, Dynamic Link Library), DLL 文件是平台中功能模块的最小单元,单个 DLL 文件可以具有复杂的功能,如状态监测模块,一个 DLL 文件里既包含了机组树形列表、监测内容分类、监测子界面分类,又包含了主监视图、数据棒图、实时趋势等多种监测界面.这样平台框架只需要包含 1 个 DLL 文件,就可以封装为独立的水电机组状态监测系统,与相应的机旁采集单元结合后,进行打包和商业应用.

对于部分可重用的功能,则可细化为多个 DLL 文件,如历史数据的趋势分析、相关分析,波形数据的频谱分析,甚至于使用较多的测点选择界面,都可以开发为独立的 DLL 文件,在框架需要时加载.如趋势分析前选择测点时,可先调用测点选择的 DLL 文件,选择的测点通过内存共享的方式,传递给趋势分析的 DLL 文件.这样,平台框架通过打包数个趋势分析相关的 DLL 文件,就可以封装为独立的水电机组趋势分析系统,主要针对历史数据进行数据挖掘和分析.

平台框架通过封装不同 DLL 文件形成不同独立应用系统的结构示意图如图 2 所示.

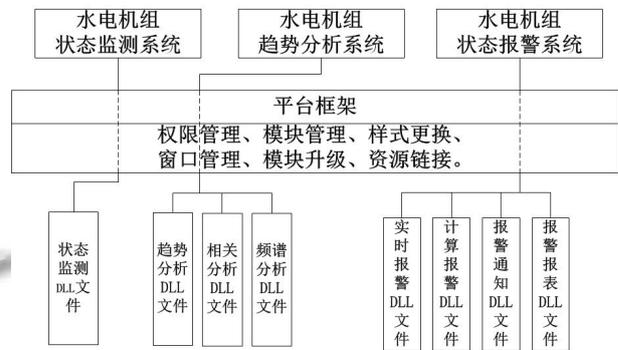


图 2 封装 DLL 文件形成不同应用系统

由于模块 DLL 的功能与开发技术并不一定相同,因此,为了统一调用和界面展示,需要规定统一的载入标准.以 DLL 中生成主 Form 为例,在 DLL 项目的 Project Source 中,定义 Form 生成函数:

```
function CreateDLLForm(App: TApplication; Scr: TScreen):TForm;
```

同时需要保证 DLL 文件载入时能获取宿主(平台框架)的 Handle,即采用如下语句:

```
DLLApp := Application;
```

```
DLLScr := Screen;
```

而在 DLL 文件释放时,采用 Windows 的回调函数 DLLProc,调用 ExitDLL 过程,将句柄还给平台框架,语句为:

```
DLLProc := @ExitDLL. \
```

对于部分所有 DLL 都会用到的函数,将使用标准的定义,使得所有的 DLL 文件都能正确实现动态菜单加载等功能。

如在 DLL 的 Form 中定义输出描述和菜单序号的函数:

```
procedure DescribeDLL(var Desc: string; var
FormTag: Integer); export; stdcall;
```

4 数据共享与通讯

由于系统功能模块全部编制成了 DLL 文件,而应用程序进程会把 DLL 映射到自己的地址空间中,因此,每个 DLL 文件都拥有自己的数据实例,应用程序对一个 DLL 中全局变量的修改不会影响到其他的 DLL 文件。

以趋势分析系统为例,从公用的测点选择 DLL 文件模块中,我们可以采用精确和模糊查询方式进行测点其相关配置信息的选择,选择的结果,就必须通过数据共享的方式,输出到分析模块 DLL 文件中。

这里采用的内存映射文件方式,内存映射文件通过映射一个文件中指定的区域或者指定的内存块,数据就可以通过内存只带来直接访问,既提高了文件存取速度和效率,也满足了各 DLL 模块间的数据共享和通讯要求。以最简单的用户登录信息传递为例,首先是创建内存映射:

```
ShareMemName := 'UserInfo';
```

```
FileHandle := CreateFileMapping($FFFFFFFF, nil,
PAGE_READWRITE, 0, SizeOf(ShareMem),
PChar(ShareMemName));
```

然后是映射到本地进程中:

```
UserInfoShareMem := MapViewOfFile(FileHandle,
```

```
FILE_MAP_ALL_ACCESS, 0, 0, sizeof(ShareMem));
```

然后是给共享内存赋值,由于 UserInfoShareMem 是 PShareMem 类型,也就是指向共享内存的指针,因此将对应的字段值直接赋给共享内存:

```
UserInfoShareMem.UserNo :=User_Detail_info.UserNo;
```

这样就通过共享内存的方式,实现了数据共享和通讯功能,不管是用户登录后获取的权限,还是测点选择时选择的数据点,以及分析结果与报表生成的数据传递,都可以用这种方式得到很好的解决。

5 结语

对水电机组运行设备的数据分析,主要依据数据挖掘的方法和软件,而建立状态数据的应用平台框架,并为各子模块建立编程标准,则为水力机械状态数据的应用,提供了一个开放的软件平台基础,可以让更多的人参与到数据挖掘和分析工具的开发中来,而不是之前仅仅依靠状态监测厂商的力量。

当然,水电机组状态监测技术走到今天,在数据采集和存储方面技术的确已经非常成熟,因此,提出标准化平台的思路,并从几个节点入手,实现数据存储和数据应用的标准化,既能解决状态监测系统的功能从数据采集向数据应用转变中的困难,又符合开放协作和标准化的技术路线,最终能为水力机械的故障诊断和状态检修真正提供依据和基础。

参考文献

- 1 刘晓亭,冯辅周.水电机组运行设备监测诊断技术及应用.北京:中国水利水电出版社,2010:327.
- 2 王春红,陈继红.基于 Delphi 的 DLL 封装系统数据支撑模块技术的研究.现代计算机,2012(5):51-54.
- 3 秦燕峰,卢菁,刘亚军.基于多文档和动态链接库技术的软件开发方法研究与实现.计算机应用研究,2002(1):51-53.