飞机排故专家系统的EASERVER三层结构设计

董健康[1]，耿 宏[2]

(1. 中国民航学院 天津 300300; 2. 中国民航学院机电学院 天津 300300)

【摘要】针对航空公司维修部门及其维护信息库多而分散的特点，提出了一种基于EASERVER应用服务器的C/A/S三层软件结构解决方案，该方案作为飞机排故专家系统，优化了专家系统的软件结构，避免了硬件升级，并节省了费用。采用不可视对象和DATAWINDOW结合处理技术，满足了专家系统对大量数据处理和数据共享的要求。

关 键 词 飞机排故专家系统; 软件结构优化; 三层结构; 应用服务器

中图分类号 TN914.42 文献标识码 A

A Desgin of the Three-Tier Scheme of EASERVER Application Server in the Troubleshooting Expert System

Dong Jiankang[1]，Geng Hong[2]

(1.General Administration Office,CAUC Tianjin 300300; 2.College of Aeronautical Mechanics and Avionics Engineering,CAUC Tianjin 300300)

Abstract Considering many separated maintenance databases of the airline companies, this paper presents a solution based on the C/A/S construction of EASERVER application server. The solution is applied in the troubleshooting expert system .It optimizes the software construction of the expert system, and avoids upgrade of hardware so as to reduce the cost. By combining NVO(Non Visual Object) with DATAWINDOW, it solves the problem of dealing with and sharing a lot of data in the expert system.

Key words aircraft troubleshooting expert system; optimization of software construction; three-tier scheme; application server

为实现民航各维修部门工作的集中管理、指令维修，充分利用民航飞机排故专家系统，共享维修信息，实现维修技术集成，需解决两个主要技术问题：一是采用怎样的软件结构，优化不同地域多数据库的访问；二是采用何种技术，在多用户并发访问时，提高数据处理和共享能力，满足民航飞机排故专家系统大数据量处理的要求。

1 软件结构的优化设计

航空集团重组后，访问用户数量大增，数据库服务器经常因针对每个对象实例单独建立一个连接，而引起数据连接池超载。客户端也会因连接多个数据库造成商务逻辑复杂度增加而不堪重负，造成所谓“胖”客户端。

采用C/A/S三层软件结构可解决上述问题，避免了大量的硬件升级。如图1所示，将两层结构客户端中的商务逻辑部分移到中间层(即：应用服务器)，仅把表现层逻辑留在客户端，客户端只要连接到应用服务器，即可通过应用服务器处理大量数据，从而大幅度减少客户端的逻辑复杂度，使其“瘦”下来。通过应用服务器的数据库共享连接池，共享客户端与各数据库的连接，避免了数据库服务器为每个访问对象实例单独建立一个连接而造成的数据库服务器端数据连接池超载。也避免了各维修部门服务器和客户机的硬件升级。在此方案下，只需增加一台性能较高的应用服务器和相应的软件即可，这样便节省了费用。

收稿日期：2003 – 11 – 26

作者简介：董健康(1960 –)，男，硕士，副教授，主要从事航空电子设备及微机测控系统方面的研究.

采用PB-JAGUAR CTS-DB三层软件结构，使用POWERBUILD语言，运行效率高于JAVA。特别是通过中间层(JAGUAR CTS)的不可视对象(NVO)、共享和运行商业逻辑，以及在中间层使用DATAWINDOW共享数据，满足了民航飞机排故专家系统对数据处理和共享能力的要求。查询客户端的设计，由于功能简单，本文采用JSP-POWERDYNAMO-DB三层软件结构。可使JSP浏览器客户端避免由用户安装客户端软件；另外，采用JSP编写查询页面和JAVA BEAN的查询方法，保证了用户快捷地查询排故知识和排故专家建议。

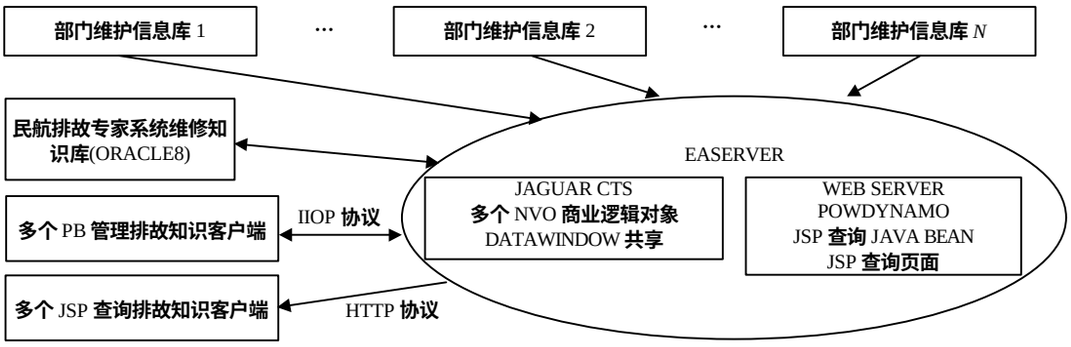

图1 民航飞机排故专家系统C/A/S三层软件结构图

2 数据处理和共享设计与实现

现以飞机排故专家系统中，数据处理和共享方面具典型意义的排故知识获取流程为例，说明数据处理和共享的设计实现。

2.1 排故知识获取流程

排故知识获取是维修工程师使用PB管理排故知识客户端，并从大量维护信息中筛选整理出排故知识，调用模糊统计推理机计算专家建议的过程。其流程如图2所示。

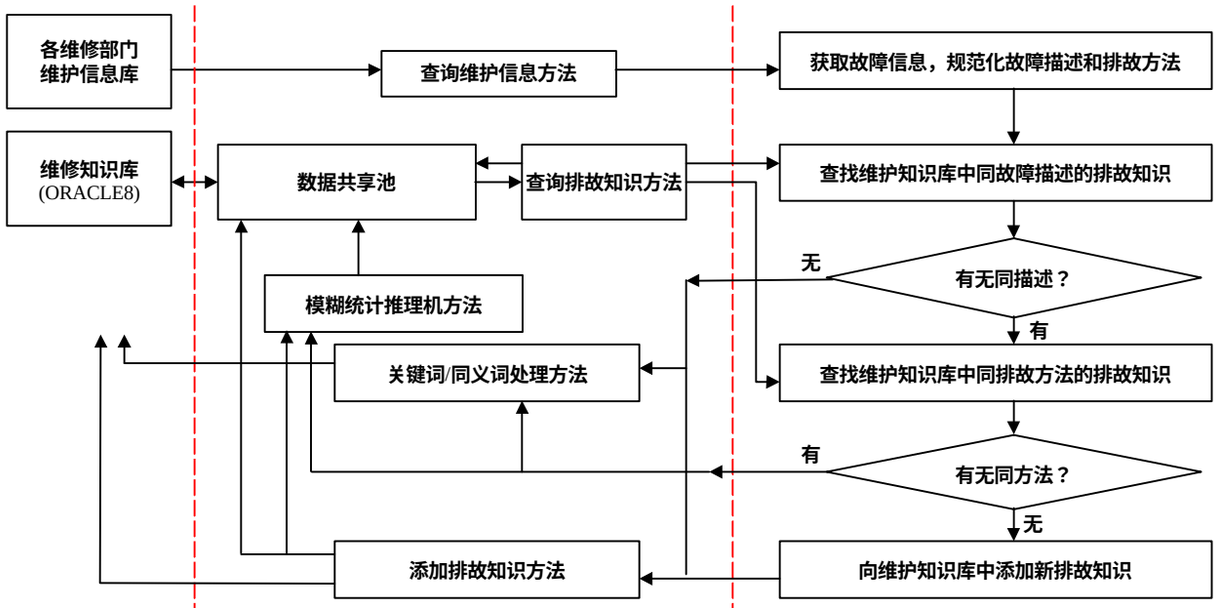

图2 C/A/S三层结构的排故知识获取流程图

排故知识获取是维修工程师与知识获取软件交互的结果。首先，客户端按故障现象从维修信息库中查询一条排故信息，规范其故障描述和排故方法。其次，调用查询排故知识方法，查询维护知识库中所有包含上述故障描述的排故知识，填充数据共享池，维护工程师从中选择相同的故障描述，若无，则调用添加排故知识方法，存储排故知识，调用关键词/同义词处理方法，提取故障描述中的关键词和同义词，存入关键词/同义词库，流程重新循环；若有，则调用查询排故知识方法查询共享数据池中该故障描述下全部排故

方法。最后，维修工程师判断当前排故信息中的排故方法与数据池中查询出的排故方法有无相同，若无，调用添加排故知识方法向共享数据池添加一条新排故知识，并调用模糊统计推理机方法计算排故专家建议，存储排故知识及其专家建议，流程重新循环；若有，调用模糊统计推理机方法计算相同排故方法使用频度加一后的排故专家建议，存储专家建议，流程重新循环。

2.2　数据处理与共享设计

将JAGUAR CTS组件应用服务器技术，与DATAWINDOW数据处理技术相结合，在应用服务器端建立DATAWINDOW共享数据池，使用JAGUAR CTS不可视对象(NVO)建立数据处理和共享的方法，客户端通过NVO代理对象(PROXY)调用应用服务器端的NVO方法，实现利用应用服务器对数据的高效处理与共享，我们称该技术为分布式PB数据处理与共享技术。

如图3所示，建立两个应用程序(stss_ClientApp和stss_ServerApp)。在stss_ServerApp中，创建类对象(stss_Class)，将商务逻辑抽象为stss_Class的类方法及相关数据窗体，建EASERVER部件工程，并将它配置到EASERVER应用服务器，使stss_Class成为NVO，stss_ServerApp成为JAGUAR CTS的应用。建实例变量(BLOB bl_share)，作为应用服务器数据共享池，实现排故知识获取流程的应用服务器化，仅仅将其表现层逻辑留给stss_ClientApp。

在stss_ClientApp中，创建EASERVER代理工程，并将其配置到EASERVER和stss_Class，建客户端连接对象(connection)的全局变量(jarConnect)和stss_Class的全局变量(iu_jarServer),前者用于连接应用服务器，后者用于代理应用服务器端的NVO和数据窗体。建立客户端窗体，实现排故知识获取的表现层逻辑。

此外，在应用服务器端有TRANSPORT对象，它监听任何用户请求，在客户端有CONNECTION对象，用来与应用服务器连接，两者通过IIOP协议通信。

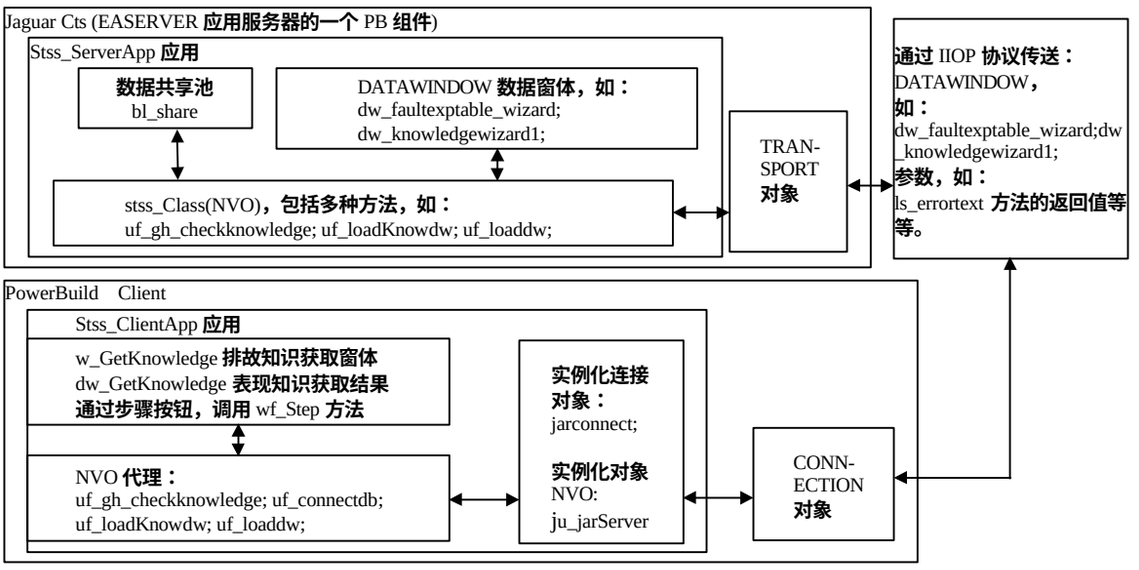

图3　针对排故知识获取流程的分布式PB数据处理与共享技术流程图

2.3　分布式PB数据处理与共享技术编程实现

2.3.1　排故知识获取客户端编程实例

根据2.1节和图3所示，排故知识获取的客户端流程通过窗体(w_GetKnowledge)来表现。在窗体中，添加DATAWINDOW(dw_GetKnowledge)，接收通过NVO处理的服务器端DATAWINDOW；添加流程向导按钮，执行流程向导方法(wf_Step(integer i_step) return integer)，其典型程序如下：

```
choose case i_step
    case 1                    //第一步：获取故障信息，规范化故障描述和排故方法
        ------------------
        this.dw_ GetKnowledge.Setfullstate(iu_jarServer.uf_loaddw("dw_faultexptable_wizard",
```

```
        ls_errortext))
    //将服务器端的数据窗体dw_faultexptable_wizard传给客户端的dw_ GetKnowledge，
    //供维修工程师选择，规范化故障描述和排故方法。
    ------------------
case 2                  //第二步：查找维护知识库中同故障描述的排故知识
    ------------------
    iu_jarServer.uf_gh_checkknowledge("",ls_keywords,"",ls_errortext)
    //代理NVO方法uf_gh_checkknowledge查找维护知识库中与ls_keywords相同故障描述
    //排故知识
    this.dw_ GetKnowledge. Setfullstate(iu_jarServer.uf_loadKnowdw("dw_knowledgewizard1",
        ls_errortext))
    //将服务器端的数据窗体dw_knowledgewizard1传给客户端的dw_ GetKnowledge，
    //供维修工程师选择同故障描述排故知识。
    ------------------
end select
```

2.3.2 应用服务器端编程实例

根据图3中的服务器端数据处理与数据共享池过程。在stss_Class对象中定义实例变量：

```
Private blob bl_share                   //作为服务器端NVO的数据共享池
```

针对2.3.1 中第二步的数据处理，客户端仅仅表现处理结果，并且服务器端的NVO数据处理过程是：1) uf_gh_checkknowledge方法查找维护知识库中与ls_keywords相同故障描述的全部排故知识，并存入bl_share；2) uf_loadKnowdw方法对bl_share进行数据整理，并将结果传给dw_knowledgewizard1和bl_share，再通过blob类型传给客户端。uf_loadKnowdw的典型程序如下：

```
ls_errortext=""
blob bl_data
datastore ds_buffer
datastore ds_data
ds_buffer=create datastore
ds_data=create datastore
ds_data.dataobject="dw_knowladgebodywizard1"
ds_buffer.setFullstate(bl_ share)
    ------------------          //对ds_buffer的处理，并将其传给ds_data
ds_buffer.GetFullState(bl_ share)   //将ds_buffer的处理结果传给bl_ share
ds_buffer.GetFullState(bl_data)     //将ds_buffer的处理结果传给bl_ data，以便输出
    ------------------          //错误处理，将结果返回给ls_errortext
destroy ds_data
destroy ds_buffer
return bl_data
```

参 考 文 献

[1] Sybase I. Powerbuild online books[M]. Seattle: Sybase Inc, 1999

[2] Richard M H. Enterprise JavaBeans[M]. 北京: 中国电力出版社, 2001

[3] Payne E C, Macarthur R C. Developing expert system[M]. New York: Wiley & Sons, 1990

[4] 王 珊. SYBASE原理-高级系统管理与性能调优[M]. 北京: 中国水利水电出版社, 1998

编 辑 孙晓丹