

基于 C++ 的飞机黑匣子解码分析系统的设计^①

Design of Aviation FDR Decode and Analysis System with C++

张蓝春 梁红光 (海口经济学院 信息工程学院 海南 海口 570203)

摘要: 本文提出了一种新的基于 C++ 的飞机黑匣子解码分析系统的设计方案。文章中分别从系统模式、后台服务器数据库系统设计、底层解码模块、数据分析与客户界面层设计等方面阐述了系统设计的具体方法。系统开发实践表明本文提出的系统设计方法是确实可行的, 采用本文提出的系统设计方法设计的 FDAS 系统填补了我国在民航飞机黑匣子解码系统方面的空白, 并荣获中国民航科技进步奖。

关键词: 黑匣子 解码器 参数 飞行数据分析 设计

1 引言

飞机黑匣子以及快速存取记录器(简称 QAR)的解码分析是中国航空公司适航运行的必要系统, 任何航空公司如果没有黑匣子或 QAR 的解码分析系统, 将不能正常运营。由于黑匣子/QAR 解码技术被国外的大公司垄断, 长期以来, 中国绝大部分航空公司都只能从国外进口相关的软件系统, 价格昂贵; 相应的采用黑匣子进行的飞行安全调查也只能依赖国外的相关公司, 中国人自己在这方面没有话语权。因此, 设计具有中国人自主知识产权的黑匣子/QAR 解码分析系统具有重要的经济价值和意义。

本文作者之一在国内大型航空公司从事黑匣子解码研究十多年, 终于在 2006 年设计出相关系统并投产, 为航空公司节省进口软件系统成本 500 多万。本文揭示了黑匣子/QAR 解码分析系统的设计理念, 并以实际的解码分析系统为背景讲述了如何具体设计黑匣子解码分析系统的方法。

2 黑匣子数据格式及其特点

飞机黑匣子的数据记录通常为一个二进制文件, 文件中的数据通过帧、子帧和超帧的方式划分。一帧分为 4 个子帧, 每个子帧为 1 秒记录的数据, 超帧指 4 帧数据(即 16 个子帧的数据)。每一个子帧的第一个数据字为同步字, 用来标志是第 1 到第 4 个字帧。目前黑匣子有两种同步标准, 分别称为 HAMILTON 同

步标准和 TELEDYNE 同步标准, HAMILTON 同步标准中第 1 到第 4 子帧的同步字分别如下(16 进制): E24, 1DA, E25, 1DB; TELEDYNE 同步标准中第 1 到第 4 子帧同步字分别如下(16 进制): 247, 5B8, A47, DB8; 通过对黑匣子数据的位运算获取同步字的位置, 就可以将二进制的原始数据有序划分开来, 为进一步的解码做好准备。

3 系统模式

考虑到大规模的飞行数据解码分析处理的需要, 因此, 系统采用 C/S 设计模式, 即由一台服务器带多台工作站的模式, 各个工作站实现解码分析处理, 经过处理过滤后的关键性数据存储在后服务器器上。

3.1 系统部署结构

飞行数据译码分析系统是一个 C/S 应用系统, 系统部署在 Windows SQL Server 服务器上, 但大量的 QAR 原始数据保存在各个译码工作站上, 而没有将原始数据保存到服务器上。因为大量 QAR 原始数据保存在服务器会使 QAR 数据译码分析处理效率大大下降; 同时也难以长期保存所有的 QAR 数据。而分散保存在各个不同的译码工作站上则不存在上述问题。部署结构对本系统如何部署在网络环境上作了建模, 如图 1。

软件架构

^① 收稿时间:2009-02-09

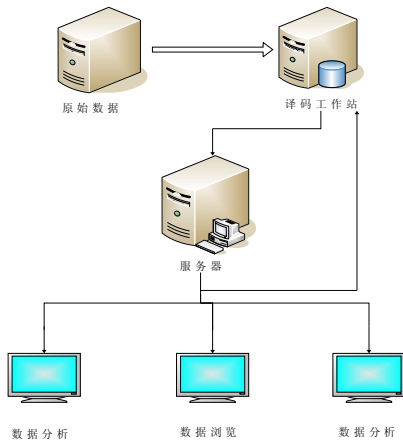


图 1 飞行数据译码分析系统软件架构图

3.2 软件架构



图 2 飞行数据译码分析系统软件架构图

4 后台服务器层的大型关系数据库与前台小型文件数据库的搭配使用

系统对黑匣子/QAR 解码分析后的所有航段关键性的索引数据将存放于后台服务器。在服务器数据库选择上应考虑数据库容量问题，因此应该选择容量较大的数据库管理系统，目前数据库行业流行的 SQLServer 和 ORACLE 两种数据库管理系统是比较理想的选择。在系统运行中还有象飞机数据库、机场数据库等容量比较小的数据库表单，采用存放于工作站当地的方式处理，这样有利于加快数据库访问速度。这些小型的数据库表单也可单独采用例如 ACCESS 之类的小型数据库存储，这样不仅维护方便，而且系统访问速度快。

后台服务器数据库对于写入、修改、删除频繁的

命令设置相应的存储过程，这样前台系统解码后的关键数据只需要交给后台服务器数据库自行处理即可，这样可大大加快系统运行速度，避免多台工作站之间为了抢占资源而导致互相锁死的现象。以下是一个具体的存储过程的 SQL 数据库语言程序例子，即查询记录的存储过程程序：

```

CREATE procedure pro_InquireEvent
(
    @registration varchar(10) = '%',
    @flight_date varchar(10) = '%',
    @replay_date varchar(10) = '%',
    @replay_time varchar(10) = '%',
    @to_airport varchar(10) = '%',
    @ld_airport varchar(10) = '%',
    @code varchar(10) = '%',
    @occur_time varchar(10) = '%'
)with encryption
As
Begin
    if exists (SELECT* FROM [FOQA].[dbo].[EVENT]
        WHERE (registration LIKE @registration)
        AND (flight_date LIKE @flight_date)
        AND (replay_date LIKE @replay_date)
        AND (replay_time LIKE @replay_time)
        AND (to_airport LIKE @to_airport)
        AND (ld_airport LIKE @ld_airport)
        AND (code LIKE @code)
        AND (occur_time LIKE @occur_time) )
        return '1'
    else return '0'
end
GO
    
```

其中，registration 字段表示飞机注册号，flight_date 字段表示飞机飞行日期，replay_date 字段表示系统解码分析日期，replay_time 表示系统解码分析时间，to_airport 字段表示飞机起飞机场，ld_airport 字段表示飞机着陆机场。该存储过程表示首先接受程序发送过去的 registration、flight_date 等字段值，然后查询数据库中是否有一样的记录，如有则返回 1，否则返回 0。

5 系统底层解码器设计

系统底层解码器需要直接处理编码后的飞机黑匣子/QAR 原始数据文件，并实现解码，要求系统运行速度快而且平稳。因此，这一部分最好采用 C++ 进行程序设计，这样有利于系统快速而平稳地运行。由于飞机黑匣子的每个数据字是采用 12 个位的方式进行编码，而计算机字节只有 8 个位，为此，在读写黑匣子字节的时候要进行相应的转换。对于具体参数的解码方式，通常分为开关量参数、多项式参数、离散参数、多段离散参数、多段模拟参数、BCD 编码参数等等。将参数分类并设计针对每类参数的相应的解码函数，以便解码器调用。对于非线性曲线应该通过插值或差分法计算其相关的工程值，进行曲线拟合；采用飞机制造厂家提供的数值点，对整个曲线进行插值拟合；对厂家没有提供数值点的数值区间，而仅仅提供数值边界条件的情况下应该通过差分算法计算曲线各点的工程值。根据参数具体情况可采用前差、中差、后差等算法：

$$\text{前差: } \delta_x^{\rightarrow} = \frac{u_{i+1,k} - u_{i,k}}{\Delta x}$$

$$\text{后差: } \delta_x^{\leftarrow} = \frac{u_{i,k} - u_{i-1,k}}{\Delta x}$$

$$\text{中差: } \delta_x = \frac{u_{i+1,k} - u_{i-1,k}}{2\Delta x}$$

6 飞行数据分析层设计

通过解码器解码后的飞行参数数据将传送到数据分析层。通过数据分析来发现飞机运行的安全隐患。例如，通过对飞机操纵数据的分析，来发现飞行员操纵飞机的技术缺陷，通过发动机参数分析来找出飞机故障并排除故障。在数据分析中较常采用的分析方法有积分法、微分法等。例如：对飞机接地距离过长的情况监控算法，该算法核心是：从飞机进跑道开始对飞机的位移进行微分处理，并投影到水平地面上，然后进行积分算出飞机接地点距离跑道端点的距离，从而判断飞机接地过程中是否平飘距离过长。根据积分方程

$$\int_{\text{height} = 50 \text{ feet}}^{\text{空地电门}} v dt \text{ 设计以下具体的分析程序:}$$

```
if(air && (ph==7 || ph==6) && swt704a)
{
```

```
    if(swt704)count704++;
    count704a=Gs*1.852*1000/3600+count704a;
}
```

其中，Gs 是飞机地速，根据 $\Delta s=v*\Delta t$ 进行积分就可得到飞机进跑道开始的平飘距离。swt704a 是一个开关变量，表示在何种逻辑设置下，系统开始对飞机接地距离长的问题进行监控，air 表示飞机在地面还是空中，ph==7 表示飞机正在着陆阶段，ph==6 则表示飞机处于进近阶段。

7 界面显示层设计

界面显示采用常用的 Windows 界面组件显示数据，包括常用的网格组件、文件显示管理组件、进度条、工具栏、弹出式菜单等等。底层的原始数据通过解码器解码后发送到界面层，形成用户能看懂的工程数据。系统数据显示界面设计可采用 C++BUILDER、DELPHI、VB 等 RAD 开发工具。

用户对界面数据的操纵命令传送到原始数据底层处理，对于数据量大的 QAR 警告记录处理命令，如果直接在界面上对网格记录组件内的数据进行排序操作等处理的话，将导致处理速度非常慢，不能满足实际工作要求。因此，将用户命令传送到 C++ 数据处理底层，在系统底层采用 C++ 的 STL 进行排序、筛选等处理，然后将根据用户命令处理后的数据显示在界面层上，这样可大大加快用户命令反应速度。将解码后的数据输出到 3D 仿真软件，对飞行过程进行 3D 动画模拟。使用开源的 flightgear 软件作为 3D 动画软件原型，然后加上数据驱动功能就形成了我们所需要的 3D 动画模拟软件。图 3 是具体的软件界面插图：

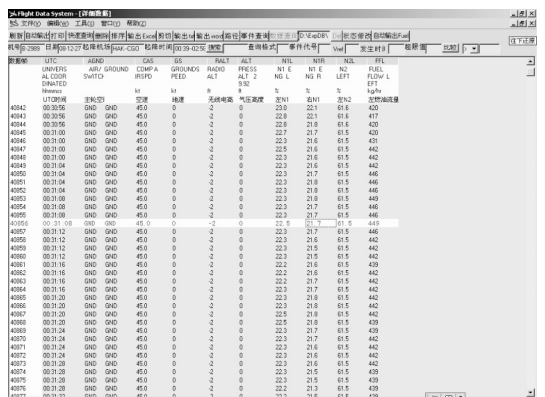


图 3 QAR 解码后的工程数据显示界面

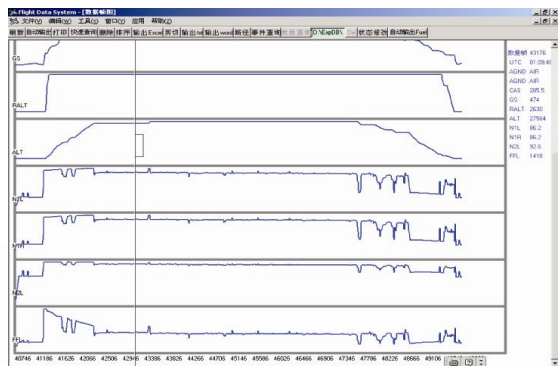


图4 飞行参数的曲线显示方式



图5 飞机飞行的三维动画模拟

8 结论

作者采用本文提出的黑匣子解码分析系统设计方法,在国内大型航空公司成功地设计了相应的飞机黑匣子解码分析软件:FDAS 软件系统。该软件系统填补了国内在这方面的空白,通过了中国民航总局的科技成果鉴定。

参考文献

- 1 Schildt H. The Complete Reference C, the Fourth Edition.
- 2 Schildt H, Guntle G. Borland C++ Builder: The Complete Reference.
- 3 The Boeing Company: Digital Flight Data Acquisition Unit 737-300/400/500 Database Interface Control and Requirements. Document No. D226A101-1.
- 4 Hollingworth J, Butterfield B, Allsop BJ: C++Builder5 Developer's Guide.
- 5 Wright RS, Jr. Lipchak B. OpenGL SuperBible, Third Edition.