

DOI: 10.19416/j.cnki.1674-9804.2019.02.002

基于 WPF 的遥测 CAS 信息实时监控 系统设计开发

Design and Development of Telemetry CAS Information Real-time Monitoring System Based on WPF

冯 灿 刘 涛 刘亚妍 / FENG Can LIU Tao LIU Yayan
(中国商飞民用飞机试飞中心, 上海 201323)
(COMAC Flight Test Center, Shanghai 201323, China)

摘 要:

在当前试飞模式下,立足于某型客机机组告警系统(Crew Alerting System,简称 CAS)信息复杂、重构平台单一等不足,设计了一套 CAS 信息实时监控系统。系统基于 WPF(Windows Presentation Foundation,简称 WPF)框架,采用数据访问层、业务逻辑层和显示层组合的典型三层架构,以遥测脉冲编码调制(Pulse Code Modulation,以下简称 PCM)数据为输入,结合 DataBinding 数据驱动模型和 XAML 等技术实现了 CAS 信息的实时解析、传输、分发及显示。同时,极大地降低了系统间的耦合性,提高了实时性能和系统的可维护性。通过滑行、首飞等试验的充分验证,系统实现了该型客机 CAS 信息的 100% 地面复现,为后续民机试飞测试、监控技术的创新提供了参考依据。

关键词: CAS; WPF; 三层架构; DataBinding; XAML; PCM

中图分类号: V217

文献标识码: A

OSID:



[Abstract] In the current flight test mode, based on the lack of CAS information of a passenger aircraft and the single reconstruction platform, a set of CAS information real-time monitoring system was designed. Based on the WPF framework, the system uses telemetry PCM data as input, and uses a typical three-layer architecture, combined with DataBinding model and XAML technology to realize real-time parsing, transmission, distribution and display of CAS information. At the same time, the system greatly reduces the coupling between systems, improving real-time performance and system maintainability. Through the full verification of the tests such as taxiing and first flight, the system realized 100% ground recurrence of CAS information of this type of passenger aircraft, which could provide a reference for the subsequent innovation of flight test and monitoring technology of civil aircraft.

[Keywords] CAS (Crew Alerting System); WPF (Windows Presentation Foundation); three-tier architecture; DataBinding; XAML; PCM

0 引言

遥测数据实时监控系统是民机试飞过程中不可或缺的重要系统,同时也是面向试飞工程师、指挥人员以及飞机设计人员的支持性系统,其主要功能是

利用遥测数据驱动可视化显示控件,实时反馈飞机的飞行状态以及各系统性能指标甚至是飞行试验结果等信息,从而为监控人员提供决策性依据^[1]。现代大型民机大量使用先进技术,系统多而复杂,其中,机组告警系统(Crew Alerting System,简称 CAS)

[基金项目] 大型民机试飞数据高速实时处理及异常状态检测分析平台(17511105000)。

作为飞机的健康管理中枢,汇聚了来自飞机各个系统的关键状态,因此,在地面监控中利用遥测数据重构飞机 CAS 系统对于增强试飞安全、提高试飞效率、改进试飞策略等具有重要意义。

国内主流试飞机构的监控程序的开发主要依赖于 Labview、VC ++、DataView 等平台^[2],在功能实现、效果显示、系统维护等方面都具有较大的制约性,更难以满足大型民机多参数、多终端、高实时的监控需求。本文立足于 WPF 平台,以实时性、稳定性以及可维护性为指导,利用三层架构模式设计开发某型客机遥测 CAS 信息实时监控系统。WPF 是 Windows Presentation Foundation 的简称,是微软推出的基于 Windows 的用户框架^[3]。WPF 集成了矢量引擎,配合三层架构在 CAS 监控应用的开发方面具有以下优势:

WPF 提供了丰富的 UI 表现形式,大大增强了人机交互界面的显示效果,同时以 XAML 文件的方式设计、维护程序界面,最大限度的降低了 UI 显示和逻辑实现之间的耦合程度;DataBinding 模型简化了数据驱动模式和线程调度,增加了系统的实时性能;分层结构可轻易实现系统的维护和移植。

1 系统结构

遥测 CAS 信息实时监控系统采用图 1 所示的三层架构。整个架构以千兆以太网为基础,实现 CAS 参数的解析、再封装、分发和显示,从功能实现上可分为数据访问层、业务逻辑层和显示层。数据访问层由遥测数据实时处理服务器和数据服务器组成,业务逻辑层以应用服务器为核心,显示层则是各个实时监控终端以及运行于其上的 CAS 监控应用。

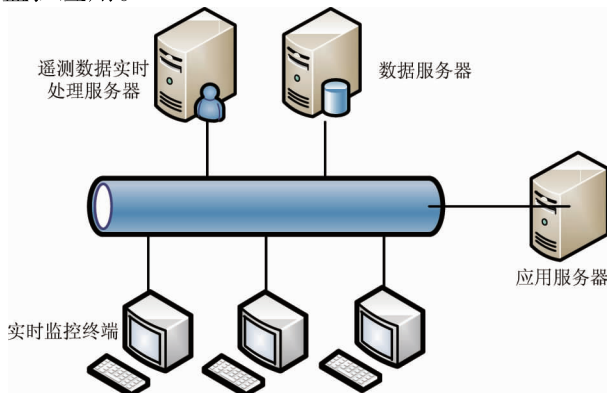


图 1 系统结构示意图

根据工作数据流向,系统以 IRIG 106 Ch10 网络数据包为输入,在数据访问层的遥测数据实时处理服务器中实现 CAS 参数的提取和相应参数值的解析,并通过数据服务器制定 CAS 信息在全网中流转的协议;业务逻辑层则按照既定协议封装和分发数据;最后,显示层根据 CAS 参数的值显示指定的告警信息。

2 系统实现

2.1 数据访问层

2.1.1 CAS 信息的遥测编码和解析

CAS 系统是在飞机出现故障时,为机组给出告警信息。某型客机的 CAS 信息驻留在机载 ARINC664 总线中,其数据类型是连续 4 000 位(这里记为 0 ~ 3 999,MSB 记为 3 999)的数据块,每一个告警信息用 Alert ID 进行唯一标识,包含 2 个数据位(bit),拥有四种状态,分别为 00 (Inactive)、01 (Active/Uninhibited)、10 (Active/Inhibited)、11 (Not used),只有当 CAS 参数的值为 01 或 10 时,才触发告警。根据告警或故障的严重程度将告警级别分为 Warning、Caution、Advisory、Status 四种,相应的告警信息在显示时也呈现出红色、琥珀色、青色和白色。为了方便解析,在遥测编码时将 CAS 数据块用连续的 250 个 PCM Word 表示。如图 2 所示,P(x,y) 作为一个 16 位的 PCM Word (其中 x 表示帧号,即 SFID,y 表示字号),代表 8 个 CAS 参数,采用“Alert_”前缀以及 ID 的组合方式对 PCM Word 进行命名,即 P(1,1) 是 PCM 主帧中第一子帧的第一个 Word,取 CAS 数据块中的 3 984 ~ 3 999 的 16 位数据,表示告警 ID 为 1 至 8 号的 CAS 参数,命名为“Alert_1_8”。

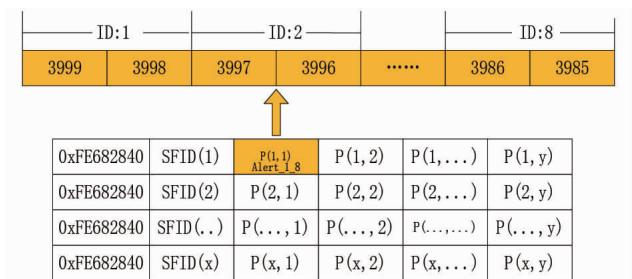


图 2 CAS 信息的 PCM 编码格式

遥测数据实时处理服务器主要完成 CAS 参数的提取和解析,工作流程如图 3 所示。核心是基于 IRIG 106 第十章网络数据包格式解析出 PCM 数据

流,并通过 PCM 同步字、SFID 等信息提取所需要的 P(x,y),最后根据预设信息提取指定的数据位,多以已经触发的告警参数形成以 Alert ID 为 key、参数值为 value 的键值对。

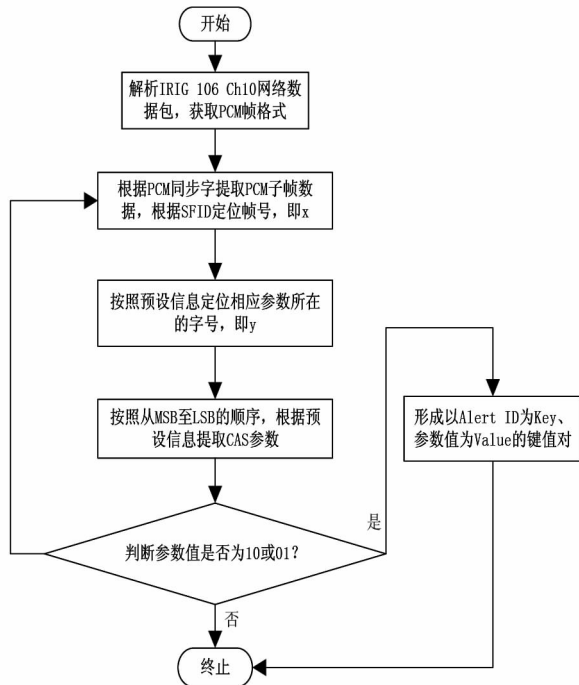


图 3 遥测 CAS 参数解析流程

2.1.2 数据服务器

数据服务器维护的 CAS 信息库是一张以 Alert ID 为主键的表,如表 1 所示。该表建立了 Alert ID 与告警内容以及与告警级别的映射关系。

表 1 CAS 信息表

Alert ID	Alert Message	Classification
17	L BLEED FAULT	Caution
21	L BLEED OFF	Advisory
276	R ENG FIRE	Warning
...
570	L NAI MAN ON	Status
...

在整个系统运行时,数据服务器建立 HTTP 服务,允许显示层的各个监控终端以 GET 的方式通过 JSON 格式获取数据服务器上的 CAS 信息表如图 4 所示,同时“指导”业务逻辑层对数据进行封装。

```

{
  CASInfor:[
    {"Alert ID":17, AlertMessage:"L BLEED
    FAULT","Classification":Caution},
    {"Alert ID":21, AlertMessage:"L BLEED
    OFF", "Classification":Advisory},
    .....
    {"Alert ID":276, AlertMessage:"R ENG
    FIRE", "Classification":Advisory},
  ]
}
    
```

图 4 遥测 CAS 参数解析流程

2.2 业务逻辑层

业务逻辑层的应用处理服务器从遥测数据实时处理服务器订阅 CAS 参数数据,同时封装网络数据包,实现面向多个监控终端的数据转发,最大程度降低了实时处理服务器的压力,提高了整个系统的稳定性,其过程如下:(1)以 GET 方式从数据服务器获取 CAS 监控参数列表,并以此向遥测数据实时处理服务器订阅相关参数,获取已经触发告警的 CAS 参数的值;(2)按照数据服务器定制的应用层协议对 CAS 参数和参数值进行封装,并通过 UDP 广播发送至各个监控终端。

封装采用 Alert ID 和参数实时值组合的方式,如图 5 所示。每个组合占用 8 个字节,0~3 字节表示 ID,4~7 字节表示所对应的实时状态值,编码均采用 ASCII 编码下的单精度浮点数,即触发告警的 CAS 参数值编码为 1.0 或 2.0。

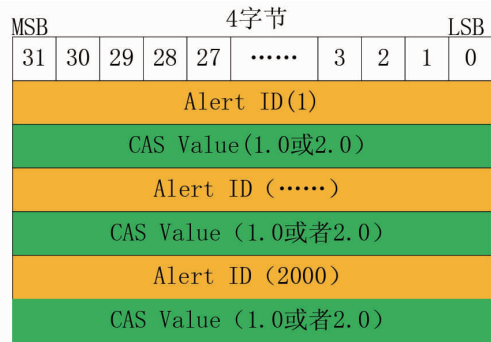


图 5 UDP 广播应用层协议

2.3 显示层

显示层是运行于各个实时监控终端的 CAS 监控应用,人机交互几面主要通过 WPF 的矢量绘制引擎实现,依靠 XAML 文件进行保存和维护。XAML

作为一种扩展文本,是实例化 .NET 对象的标记语言,定义了 WPF 应用和程序中组成窗口的面板以及各种控件的布局^[5]。为了使 CAS 监控应用适应不同的分辨率,使用 Grid 作为布局控件,将整个窗口划分为控制区、静态显示区和实时刷新区三个区域。

```

<Grid>
  <Grid.RowDefinitions>
    <RowDefinition Height="5*"></RowDefinition>
    <RowDefinition Height="10*"></RowDefinition>
    <RowDefinition Height="50*"></RowDefinition>
  </Grid.RowDefinitions>
  <Grid Grid.Row="2">
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="1*"></ColumnDefinition>
      <ColumnDefinition Width="1*"></ColumnDefinition>
      <ColumnDefinition Width="1*"></ColumnDefinition>
      <ColumnDefinition Width="1*"></ColumnDefinition>
    </Grid.ColumnDefinitions>
  </Grid>
</Grid>

```

图 6 窗口布局代码片段

在控制区利用 StackPanel 横向放置三个按钮,并依次添加最小化、正常和退出等事件,实现整个应用的控制;静态显示区用文本和颜色划分不同的告警级别区域;实时刷新区使用 ListView 控件作为 CAS 告警信息的主要载体,整个显示界面布局如图 7 所示。



图 7 CAS 信息显示界面

系统运行时,显示层的各个终端以 GET 方式请求数据服务器,获取 JSON 信息,形成以 Alert ID 为

键,Alert Message 以及 Classification 为值的两个字典:Dictionary_AlertMessage、Dictionary_AlertClassification。当显示层应用从网络数据包中提取 CAS 的参数名称(Alert ID)和值后,可通过对以上两个字典的轮询实现和告警信息以及告警级别的映射,其过程可通过图 8 的伪代码表示。

```

function CASMapping(KeyValuePairs(alert_id,casvalue))
  For item in Dictionary_AlertMessage:
    If item.key == alert_id then:
      alertMessage ← item.value
      For child in Dictionary_AlertClassification:
        If child.key == alert_id then:
          alertClassification ← child.value
        Else:
          Continue
      End If
    End For
  Else:
    Continue
  End If
End For

```

图 8 CAS 参数映射过程伪代码

用图 8 的过程的返回值“alertMessage”和“alertClassification”即可在合适的区域实时显示告警信息。信息的显示是通过 DataBinding 驱动模型实现的,如图 9 表示^[5],Binding 的数据源是遥测 CAS 参数的值,Binding 的目标就是 ListView 控件。通过 Binding,系统为每一个遥测 CAS 参数包装成为属性,并在其 set 方法中激发 PropertyChanged 事件,从而实现 INotifyPropertyChanged 这个接口,Binding 自动侦听来自该接口的事件,当目标(ListView 控件)订阅该事件时,CAS 参数的值(映射为告警信息)会在控件上更新显示。

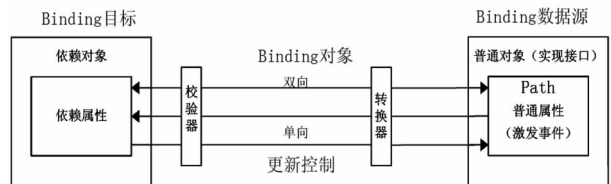


图 9 DataBinding 数据驱动模型

3 试验

为了验证本文系统的性能,设计如下对比实验:

1) 基于 Labview 开发一套独立 CAS 监控系统, 将数据接收、解析和显示集成在单一进程中, 与本文所设计的系统同时运行, 测量系统延时、CAS 准确性等指标;

2) 更新 CAS 信息库, 对比两种模式下的系统维护的难易程度;

3) 将系统移植至机载实时监控系系统, 对比系统的可移植性。

两种不同的遥测 CAS 信息实时监控系系统均以脉冲编码调制(Pulse Code Modulation, 简称 PCM)数据流为输入, 系统运行状态和性能见表 2。

表 2 试验结果

测试项目	本文系统	Labview 系统
参数数量	2 000	2 000
系统延时	< 50ms	150ms
信息准确性	100% 准确	100% 准确
终端数量	无限制	基于前端系统 Tcp 队列长度
可维护性	更新配置文件	必须代码重构
可移植性	重构数据访问层	系统重构

通过试验对比, 本文所提出的基于三层架构的 CAS 信息实时监控系系统在实时性、可维护性以及可移植性等方面更为出色: 当航电系系统发生版本迭代或更新时, 只需更新数据访问层的配置文件即可快速完成整个系系统的更新而无需重构任何代码; 在系系统移植方面, 各层级的重新编译将使得系系统很容易与不同的前端系系统融合。

4 结论与展望

本文基于 WPF 平台, 利用三层架构结合

DataBinding、XAML 等关键技术, 完成了遥测 CAS 信息监控画面的编辑、数据驱动及告警信息的实时显示, 验证了 WPF 矢量技术在试飞数据实时监控中应用, 实现了国产大型客机 CAS 信息的地面复现及实时监控, 为试飞数据监控模式及技术的创新提供了参考依据。在后续的研究中, 以该系系统为基础, 在业务逻辑层通过实时记录 CAS 信息结果, 实时侦听和响应用户操作, 实现监控过程的管理, 可为监控人员提供数据回放服务, 方便对闪现故障信息的查询和确认。

参考文献:

- [1] 白效贤, 杨廷梧, 袁炳南. 航空飞行试验遥测技术发展趋势与对策[J]. 测控技术, 2010, 29(11): 6-9.
- [2] 祁春, 段宝元. 遥测数据实时处理软件系系统及其应用[J]. 科学技术与工程, 2010, 10(28): 7047-7050.
- [3] 刘涛. 基于 WPF 技术的试飞数据实时监控系系统设计研究[J]. 中国飞行试验研究院, 中国测试, 2017, 43(1): 136-137.
- [4] Matthew MacDonald. WPF 编程宝典[M]. 王德才, 译. 北京: 清华大学出版社, 2013: 17-25.
- [5] 刘铁猛. 深入浅出 WPF[M]. 北京: 中国水利水电出版社, 2010: 35-50.

作者简介

冯 灿 男, 博士, 高级工程师。主要研究方向: 试飞与测试。E-mail: fengcan@comac.cc

刘 涛 男, 硕士, 工程师, 通讯作者。主要研究方向: 试飞数据实时处理。E-mail: liutao2@comac.cc

刘亚妍 女, 硕士, 工程师。主要研究方向: 视频采集及测试。E-mail: liuyayan@comac.cc