

测发控软件组态开发平台的设计与应用

韩亮, 张倩, 虞洋

(北京宇航系统工程研究所, 北京 100076)

摘要: 通过配置和定制式开发, 为了适应并满足快速构建运载火箭测发控软件的业务需求, 提出一种以实时内存数据库为核心, 多个软件构件为基础的组态式软件开发平台; 为设计人员提供了统一的开发接口和应用模式, 可根据具体需求选择不同的构件和模块进行组态配置和定制式开发; 目前在该平台基础上开发的数十项测发控软件已经成功应用在多个运载火箭型号之中, 证明其灵活的结构设计和高复用构件能够很好地满足需求的快速响应, 极大提高了测发控软件的实时数据服务与应用水平。

关键词: 软件开发平台; 构件; 测发控软件

Design and Application of Software Development Platform for Test Launch and Control System

Han Liang, Zhang Qian, Yu Yang

(Beijing Institute of Aerospace Systems Engineering, Beijing 100076, China)

Abstract: In order to meet the quick development need of test launch and control system in launch vehicle, the topic uses a typical design method to realize a software development platform which based on a real-time database and several software configuration items. It provides an effective application mode and unified development interface. According to the needs, developers can choose different software configuration items for customization and development. The software platform at present has been applied to Test Launch and Control System in launch vehicle, it shows that the flexible structure and high reuse items meet quick development need and greatly improve the level of real-time data services and application.

Keywords: software development platform; software configuration item; test launch and control software

0 引言

随着计算机软件技术的不断发展, 在全球范围内, 软件开发的组态化、平台化设计已经成为发展趋势。组态软件开发平台本身也是一套软件, 但不是最终与业务相关的应用软件, 而是用于二次开发的产品系列, 它是在测发控领域的需求范围之内, 封装了必须具备的一些基本功能和执行逻辑的软件框架。这个框架式和具体的业务无关的, 仅仅定义了一些为实现这些业务而必须具备的接口, 并通过这些接口搭建起来的一个完整的、可运行的软件框架。在这个平台之上, 可以根据不同的需求, 对平台定义的接口进行实现, 来实现具体的业务。

本文介绍了一种适用于运载火箭测发控领域的软件定制开发平台, 开发人员不需要编写大量的代码, 使其从繁杂的开发工作中解放出来, 把主要精力集中到业务相关功能的开发上。再通过简单的构件配置、参数录入、用拖拽的方法设计界面等操作, 就可以完成最终业务软件的开发。

1 平台组成

如图 1 所示, 组态软件开发平台由应用类、服务类、工具类和第三方产品共四类构件组成:

1) 应用类构件: 测发控软件(在组态软件开发平台上定制开发的、最终与业务相关的测发控类应用软件)在运行时所

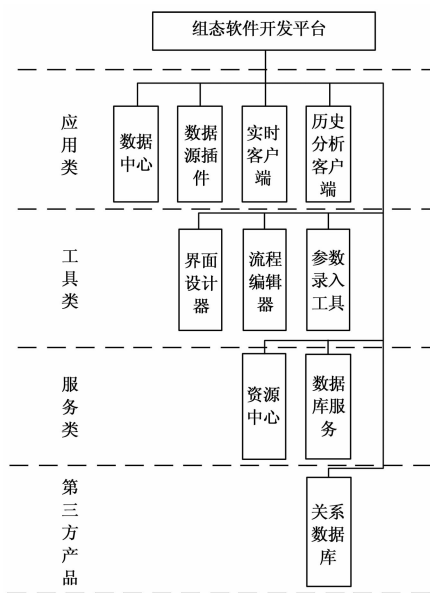


图 1 组态软件开发平台构件组成

需要的软件构件/模块。包括数据中心、数据源插件、实时客户端、历史分析客户端。

2) 工具类构件: 在组态软件开发平台上定制、开发测发控软件时, 由开发人员使用的设计器或开发工具。包括界面设计器、流程编辑器、参数录入工具。

收稿日期: 2015-01-26; 修回日期: 2014-03-13。

作者简介: 韩亮(1982-), 男, 辽宁锦州人, 主要从事综合测控、数据应用及计算机软件方向的研究。

3) 服务类构件：测发控软件在运行时或定制开发时，所依赖的、为应用类构件提供某种特定服务的软件构件。包括资源中心、数据库服务。

4) 第三方产品：即第三方的商用软件，组态软件开发平台使用的第三方产品是业界主流的关系数据库，如 SQL 或 Oracle。

2 平台设计与关键技术

2.1 构件设计

组态软件开发平台由以下几个核心构件组成：数据中心、数据源插件、实时客户端、历史分析客户端、2 个服务类构件和 3 个平台开发工具。各构件的设计及功能如下：

1) 数据中心 (DataCenter, 简称 DC) 是整个软件平台的中心，定义了测发控软件的参数全集，并对实时测试中的数据进行存储和管理^[1]。

数据中心构件的核心为两种内存数据库：即时数据库和实时数据库，即时数据库存取当前最新状态数据，而实时数据库则基于 AWE 技术对内存进行扩展，以满足测试全周期大数据的实时存储。开发人员可以根据需求选择任意一种进行挂接。

2) 数据源插件 (DataSource 简称 DS) 负责测发控软件与外系统或前端测控设备的数据通信，主要是对通信协议或底层 API 驱动进行封装，每个数据源插件都为单独的动态链接库，为数据中心加载和调用。

3) 实时客户端 (RealtimeTerminal, 简称 RT) 负责人机交互界面的图形组态化定制、客户端数据通信和人机交互的接口。实时客户端可以与数据中心集中配置，也可分布配置，可以选择发布为 C/S 架构的 exe 程序，也可以选择发布为 B/S 架构的网站，同时提供用户在线编辑功能，用户可以较灵活的设计个性界面^[2]。

实时客户端设计了轻量级的实时 workflow 引擎，内置了大量的表示流程执行过程中的节点行为：命令行为、延迟行为、等待条件行为、等待命令行为、执行脚本行为等一系列行为组件，可以满足测发控领域大多数自动流程的配置需求。自动流程引擎启动后将加载流程文件，并解析文件中开发人员预定义的的行为表达式，从而执行相应的逻辑动作。

4) 历史分析客户端 (AnalyzeTerminal, 简称 AT) 负责用户对事后数据分析与判读，包括试验数据及故障信息浏览、测试数据计算、多次试验数据的对比、多次试验数据的包络分析、输出数据文件等功能。与实时客户端类似采用了相同的技术实现，只是其数据来源是从硬盘中的关系数据库中获取，如微软的 SQL Server 数据库。

5) 两个服务类构件：资源中心 (ResourceCenter, 简称 RC) 负责对用户的界面文件、图片文件、音频文件以及软件系统所需构件的动态链接库、参数文件等进行存储管理，并提供上传下载服务；数据库服务 (DataBaseServer, 简称 DBS) 为开发平台的其他构件提供访问关系数据库的统一接口。

6) 3 个平台开发工具：参数录入工具 (ParamEdit, 简称 PE) 为测发控软件提供了可视化的参数配置工具；界面设计器 (UIEdit, 简称 UIE) 则提供了组态化的人机界面设计手段，开发人员可以通过拖、拽 (控件) 的方式，快速构建所需要的图形界面；流程编辑器 (WorkflowEdit, 简称 WFE) 为

配置测控自动流程使用。

2.2 接口设计

2.2.1 DS 与 DC 的交互

组态软件开发平台通过“数据源插件”构件将通信协议或设备驱动进行封装，并按照数据中心提供的标准接口实现，形成专用的 dll 动态链接库。数据中心作为插件容器存在，系统运行时根据配置查找和装载插件，能够动态的把插件插入数据或从框架中取出，实现软件功能的灵活配置^[3]。

以图 2 为例，两个典型的数据源插件结构：UDP 数据源插件和 PLC 数据源插件，一个封装了 UDP 的网络应用协议，一个封装了 PXI 底层驱动函数，由一个模块或是一个简单的类便可完成。两个数据源插件各自都有一个继承于抽象基类的具体实现类，实现了测发控软件向外输出控制指令的方法，这些方法被数据中心所调用，利用 C++ 多态特性从而去操作某个具体定义的数据源，完成数据的发送过程。另外，数据中心以函数指针的方式提供了多个全局函数，为数据源插件所调用，完成上行采集数据的入库、更新界面过程。

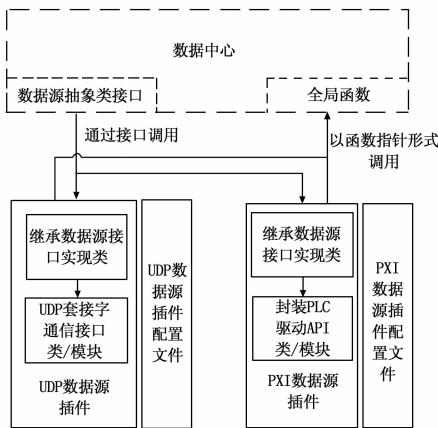


图 2 数据源插件与数据中心接口设计

2.2.2 DC 与 RT 的交互

数据中心与实时客户端信息的交互采用 TCP 的网络通信方式，因此根据需求的不同，数据中心和实时客户端可以部署在一台计算机上，或者分开部署。当登陆用户身份验证成功并下载完成该用户的界面文件之后，实时客户端向数据中心请求建立 TCP 连接，并按照平台通信协议进行交互，其流程如下：

1) 客户端向数据中心请求客户端 ID (即该客户的标识)，由数据中心生成该客户 ID，并发送给该客户端。之后该客户端向数据中心发送的每一个通信帧里都要带有此 ID 标识，以便数据中心对各个客户的身份进行辨别。

2) 实时客户端向数据中心请求本次测试基本信息：如型号、飞行批次、测试地点、测试阶段等，这些信息由数据中心发送给实时客户端，以便实时客户端进行界面的逻辑处理及显示。

3) 实时客户端向数据中心注册参数，由于登陆用户不同，浏览的界面文件也不同，因此客户端需根据界面文件上绑定的参数向数据中心注册，数据中心将维护每个实时客户端 (用户) 需要浏览的参数，从而向它们推送相应的数据。

4) 当数据中心收到实时客户端的参数注册帧后，从内存

数据库中将本次测试中这些注册参数之前缓存的数据全部提取出来, 一次性发给实时客户端。这样实时客户端即使在测试过程中途才启动, 也能看到本次测试全周期的数据。

5) 如果数据中心从数据源插件得到实时测试数据后, 立即组成实时数据帧, 主动推送给实时客户端进行显示。

6) 当实时客户端关闭时, 需要向数据中心发送注销帧, 然后断开 TCP 连接。数据中心收到注销帧后, 将取消对该客户端参数的维护管理。

2.2.3 RC 服务

资源中心是一个开放式文件发布的服务, 它拥有类似于 FTP 的文件上传下载功能, 并且具备文件的 HTTP 发布功能, 它将承担组态软件开发平台各种资源的中转和储存任务。

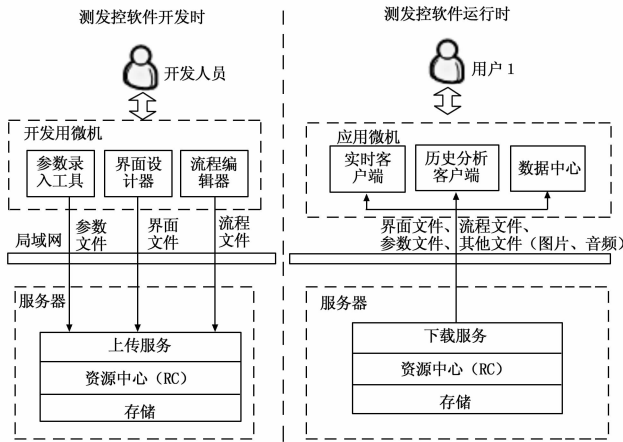


图 3 资源中心的上传下载服务

如图 3 所示, 开发人员在利用资源中心提供的上传服务, 将制作好的界面文件、流程文件、参数文件等集中保存在资源中心; 当测发控软件运行时, 每个构件去下载各自需要的文件。资源中心解决了平台复杂的文件使用逻辑, 也有利于对各种类型文件的统一管理。

2.2.4 DBS 服务

在平台的整体结构中, 关系数据库 (如 SQL Server) 中存储了系统配置信息、用户身份信息、测试数据等各种信息, 实时客户端和历史分析客户端都要获取或修改 SQL Server 中的这些信息, 因此设计了数据库服务 (DBS), 它采用微软 .Net 平台的 WCF 服务技术, 为软件平台其他构件提供统一的访问接口, 即定义了一系列访问数据库表信息的接口函数, 这些函数均封装了特定的 SQL 语句来实现各自的功能。

2.3 大规模、高实时性内存数据管理技术

为了满足数据访问的高性能以及日益增加的数据信息规模需求, 数据中心设计了实时数据库模块。它基于 Address Windowing Extensions 内存管理的扩展技术, 成功突破了 Windows32 位应用程序只能调用的 2G 内存限制。其核心原理是将申请的物理内存通过内存映射的方式映射到用户的地址空间中, 这种方案成功地解决了大容量内存的申请问题。但由于上述的程序流程必须在一个进程内完成, 因此实时数据库又设计了共享内存文件来解决多进程 (客户) 数据共享及并发问题。如图 4 所示, 内存数据库管理技术将这两者相结合方法来实现。

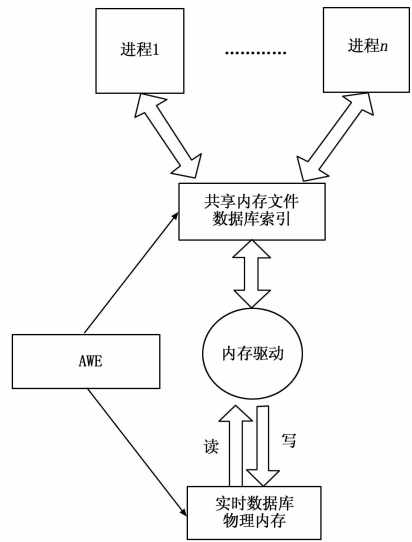


图 4 内存数据库原理示意图

AWE 技术提供的申请函数返回了一个指针数组, 这些指针数组正是物理内存的页面框号 (每个内存分页 4 k), 这样为直接访问内存页面上的数据提供了地址, 由于该组地址在用户空间里无法使用, 只能在操作系统的内核空间才能进行使用, 因此访问内核空间有两种方式: 将操作系统运行在系统模式, 或采用驱动程序进行访问。前者会为应用的运行带来严重的不可靠性, 任何应用程序的数据访问错误将会导致操作系统的崩溃; 因此采用了后一种模式, 通过采用内存数据访问驱动程序来完成对于内存数据的访问。为了保证驱动程序的可靠性, 将内存数据的访问抽象成读和写, 用精简的代码在驱动程序中实现, 而大量的内存数据库结构、索引信息等通过共享内存文件的方式来完成多用户进程的共享操作。

通过这种方式, 解决了在软件中信息汇总和数据服务功能中使用内存数据库的关键技术。通过数据访问并发测试及性能测试表明: 在 Windows XP 操作系统中可以使用的内存可达到 3 GB, Windows Server 2003 操作系统上可以使用的内存规模最大可达到 10Gb; 通过采用对于实时数据库的数据存储空间进行多次重复读或写操作, 计算总时间, 之后取平均值的方法测试出内存数据库进行一次 I/O 的时间平均时间为 20 μs 左右; 通过开发测试程序进行多进程、多线程的数据并发访问程序对于实时数据库的存储空间进行读写操作, 没有出现数据混乱的状态。

3 测发控软件的开发应用

3.1 软件的组态化开发

“组态”的概念包含“配置”、“设定”、“二次开发”等含义。本文介绍的组态软件开发平台, 用户可以利用它提供的工具、构件及方法, 通过类似“搭积木”的简单方式构建一套最适合自己的测发控软件或应用系统。主要设计内容如下:

- 1) 使用参数录入工具, 完成测发控参数全集的配置, 包括参数名称、代号、值域、数据类型等属性。
- 2) 使用界面设计器, 通过拖、拽 (控件) 的方式, 快速构建所需要的图形界面, 并完成控件与测控参数的绑定关联。

3) 使用流程编辑器完成流程文件的制作, 配置每个自动流程节点的行为及其所需要的详细参数, 如延迟时间、判断条件、发出指令等。

4) 根据目标需求, 在平台下选择合适的构件, 搭建最终的测发控软件或系统。

3.2 单机类测发控软件

组态软件开发平台的构件中, 实时客户端、数据中心和数据源插件三部分可以组成一个最基本的、用于实时监控的测发控软件; 在单机模式下, 数据中心与实时客户端通常采用本机通信方式完成数据交互, 数据中心可以采用即时数据库完成数据的管理与推送。如图 5 所示, 这种典型的部署模式可以满足测发控软件的应用需求, 如运载火箭动力测控系统后端软件、指挥控制软件等。

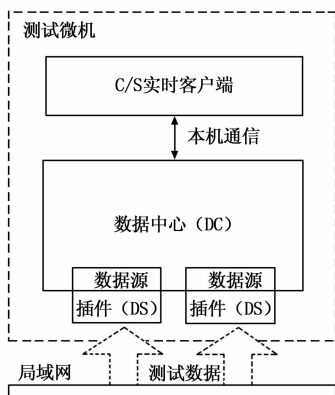


图 5 单机类测发控软件结构示意图

3.3 大型分布式软件系统

一套大型的测发控软件系统, 如运载火箭总控网系统^[4]的射前监测软件, 要求用户身份登陆、管理系统基础信息等功能, 则需要 DBS 服务配合完成对关系数据库的访问; 如果需求是一个以 B/S 的方式发布的多客户端、多用户的软件系统, 则需要 RC 的加入, 它可以对整个软件系统所有的参数文件、用户界面文件等进行统一管理, 每个用户客户端在启动后从 RC 下载所需要的文件; 如果软件系统还需要具备事后数据的分析、比对等功能, 那么就要部署历史浏览客户端来完成。

如图 6 所示, 在组态软件开发平台的框架之上, 由各个构

件配置组成了一套大型的软件系统。这些构件均为成熟的软件产品, 开发人员只需要针对测控设备或软件与外界的通信方式开发数据源插件(或使用标准、成熟的模块产品), 便可快速的搭建所需要的软件系统, 有效的控制了开发成本。

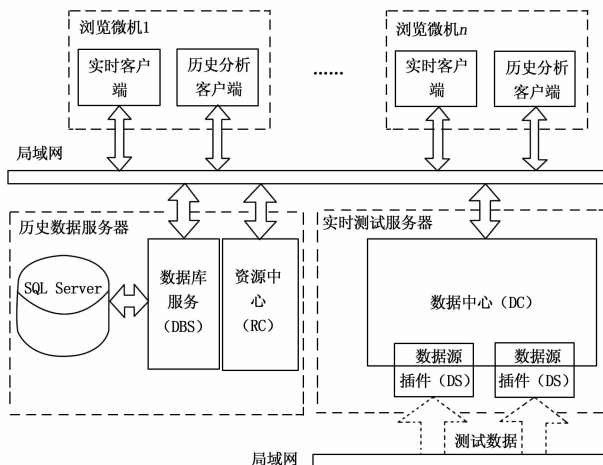


图 6 分布式测发控软件系统示意图

4 结束语

测发控软件组态化开发平台采用了信息一体化设计、内存数据库管理等先进技术, 实现了一种通用性很强的软件开发平台, 通过配置和定制式开发, 能够适应运载火箭测控领域各种软件的任务需求。具有良好的适应性、兼容性、扩展性, 为常用的测控软件系统提供了新的技术解决途径, 可以广泛应用于航空航天、工业控制及电力行业的测控领域。

参考文献:

[1] 韩亮, 汪洋, 易航. 统一测发控软件开发平台数据中心的研究与实现 [A]. 一院科技委电子组年会 [C]. 2010.
 [2] 王毅, 等. 面向监控系统的图形组态软件设计与实现 [J]. 热力发电, 2012, 41 (7): 96-100.
 [3] 陈云云, 等. 基于插件的组态软件设计与实现 [J]. 计算机测量与控制, 2013, 21 (2): 436-438.
 [4] 汪洋, 易航, 颜廷贵. 一种通用软件平台结构在运载火箭总控网系统中的应用 [A]. 一院科技委电子组年会 [C]. 2009.

(上接第 1791 页)

[2] 中国射频识别 (RFID) 技术政策白皮书 [P]. 北京: 中华人民共和国科学技术部等十五部委, 2006.
 [3] 刘伟丽. 自动跟随小车控制系统 [D]. 长春: 长春理工大学, 2013.
 [4] 赵锐, 钟榜, 朱祖礼, 等. 室内定位技术及应用综述 [J]. 电子科技, 2014 (3): 155.
 [5] 蔡磊, 周亭亭, 等. 基于超声波定位的智能跟随小车 [J]. 电子测量技术, 2013 (11): 76-79.
 [6] 纪金水. 基于 Zigbee 无线传感器网络技术的系统设计 [J]. 计算机工程与设计, 2007 (1): 404-408.
 [7] 蒋磊, 于雷, 王振肿, 等. 基于 WiFi 和 ZigBee 的井下人员无

线跟踪与定位系统的设计 [J]. 计工矿自动化, 2011 (7): 1-6.
 [8] Priyantha N, Chakraborty A, Balakrishnan H. The Cricket location support system [C]. Proceedings of MobiCom, Boston, MA, USA, Aug 2000; 32-43.
 [9] Nissanka Bodhi Priyantha. The Cricket Indoor Location System [D]. Massachusetts Institute of Technology. 2005.
 [10] 庞丽莉. 无线网络仪器的时钟同步、定位及数据安全存储算法研究 [D]. 长春: 吉林大学, 2013.
 [11] 张歌凌, 孙利娟. 基于无线传感器网络的机器人高精度定位系统 [J]. 计算机测量与控制, 2014, 22 (6): 1963-1966.
 [12] IEEE Std 802. 15. 4-2006. IEEE Standard for Information technology [S]. 2006.