

## 惯导系统测控软件通用开发平台研究

陈留涛, 杨业明, 陈京谊

(航天科工惯性技术有限公司, 北京 100074)

**摘要:** 为提高惯导系统测控软件标准化、系列化、模块化程度及开发效率, 对测控软件通用开发平台技术进行了研究。在分析惯导系统测试需求的基础上, 设计了软件平台体系结构, 并对软件平台框架的功能、组件和插件的设计、平台中的状态机模型、平台的可扩展性等进行了详细阐述。最后通过实例描述了利用软件平台开发测控软件的开发模式, 以可视化组态方式设计开发测控软件的过程表明, 软件平台具有很强的通用性, 采用软件平台开发测控软件可大幅提高测控软件的开发效率及其标准化、系列化、模块化程度。

**关键词:** 惯导系统; 测控软件; 通用开发平台; 界面组态; 流程组态

**中图分类号:** U666.12   **文献标志码:** A   **文章编号:** 2095-8110(2016)01-0073-08

### Research on Inertial Navigation System Test Software Development Platform

CHEN Liu-tao, YANG Ye-ming, CHEN Jing-yi

(Aerospace Science and Industry Inertial Technology Co., Ltd., Beijing 100074, China)

**Abstract:** In order to enhance the development efficiency and the grade of standardization, serialization and modularization of the inertial navigation system test software, a general software development platform technique of the test software is researched. The software development platform architecture is designed based on the analysis of the inertial navigation system test requirement. The function of the software development platform framework, the design method of the component and plug-in, the state machine model on the software development platform, and the function extensibility are introduced in detail. Based on a case the development model of the test software on the software development platform is described. The configuration development process shows that the general-utility of the software development platform is excellent. The development efficiency and the grade of standardization, serialization and modularization of the inertial navigation system test software based on the software development platform are advanced greatly.

**Key words:** Inertial navigation system; Test software; General software development platform; Interface configuration; Work flow configuration

### 0 引言

惯导系统是指利用惯性敏感器、方向基准和初始位置信息来确定运载体姿态、加速度、速度和位置的自主式导航系统。由于惯导系统对外通信的接口和协议多样、测试流程各异, 导致用于对其进行测试的测控软件差异很大。为满足不同

惯导产品的测试要求, 通常是针对每个产品开发专用的测控软件, 在产品研制过程中还要根据需要对测控软件源代码进行反复的修改和调试, 造成测控软件开发周期长、源代码开发量大且重复性工作多, 以及标准化、系列化、模块化程度很低, 从而制约了惯导产品的快速研发。

为解决上述问题, 本文对测控软件通用开发平

收稿日期: 2015 - 10 - 28; 修订日期: 2015 - 11 - 09。

作者简介: 陈留涛 (1979 - ), 男, 博士, 工程师, 主要从事测控软件技术方面的研究。

E-mail: clt00000@163.net

台进行了研究。测控软件通用开发平台（简称软件平台）是为测控软件开发、调试和运行提供的集成环境，可灵活配置软硬件测试资源，以可视化的方式创建测试流程，并加载硬件接口的驱动程序从而自动生成测控软件。测控软件在运行时自动载入测试过程中所需的参数、文件和数据库等，并按照设计好的流程完成对惯导产品的测试。

惯导系统对外接口及其测试流程具有多样性和复杂性的特点，因此要求软件平台要有很好的通用性、灵活性和可扩展性。为满足上述要求，本文采用组件及插件的思想设计了软件平台体系结构，并通过制定统一的组件接口规范，实现了组件的标准化开发。由于软件平台是由一系列的组件和插件构成的，组件本身所具有的可重用性和可移植性，实现了软件资源的共享和即插即用，并为软件平台功能扩展和升级提供了条件。在利用软件平台开发测控软件时，只需以可视化的方式进行一系列的设置、配置和组态设计，便可完成测控软件的开发，开发效率大幅提高。

## 1 国外及国内测控软件开发平台研究现状

### 1.1 国外典型的测控软件开发平台

PAWS<sup>[1-2]</sup>是一个综合的、开放的软件开发环境，主要用于自动测试系统软件的开发，贯穿于ATE开发的全过程，包括ATLAS程序的翻译、ATE建模、ATE驱动程序、测试程序执行等。PAWS软件平台主要包括PAWS Developers Studio、PAWS/RTS和工具软件。PAWS Developers Studio完成ATE的开发和自动测试系统资源的自动分配、PAWS/RTS完成测试程序的执行，工具软件主要是指PAWS/TRD等辅助设计ATE系统的工具。

PAWS具有层次化的体系机构，使得ATE设备开发和UUT的TPS开发可并行进行，大大缩短了开发周期。PAWS在美国军工测试领域得到了广泛应用，国内航空工业系统引进使用该软件的单位也比较多，它虽然功能强大，但需根据国内测试设备的特点和测试需求进行二次开发。

另外，国外还有一些典型的组态软件，如美国NI公司的LabView、德国西门子的WinCC和美国GE公司的Cimplicity等，利用LabView开发的测控软件在实时性方面无法满足要求，WinCC

和Cimplicity主要应用在工控领域，均不适合对惯导系统进行测试。

### 1.2 国内典型的测控软件开发平台

中航工业634所研发的GTest<sup>[3]</sup>可视化测试程序集成环境，是在对PAWS及其他国外优秀自动测试系统软件开发平台进行深入分析研究的基础上开发出来的。GTest与其他自动测试系统软件平台的区别在于：首先它使用可视化建模方式建立测试过程模型，然后结合仪器的驱动程序自动生成测试程序，不需要用户进行程序代码的编写，因此降低了对用户计算机语言应用能力的要求，有效提高了测试软件的开发效率。GTest仿照了PAWS的功能，接近实用但对其进行功能扩展和二次开发比较困难。

北京航天测控技术开发公司研发的虚拟仪器测试环境VITE<sup>[4]</sup>（Visual Instrument Test Environment），兼有测试和故障诊断的能力，功能覆盖面较广但针对性不强，若用于惯导系统测试还要进行改造。

### 1.3 研制测控软件开发平台的意义

惯导系统测控软件不但要承担所属系统的数据采集和控制功能，而且还要实现人机交互接口、数据保存与分析、系统标定与测试等功能，有时还要实现模拟仿真功能，因此功能较为复杂，实时性要求较高。另外，惯导系统在调试、验证阶段，需求变化较大，经常需要对软件进行快速、频繁更改；对惯导系统进行测控的设备通常属于定制产品，因此需要由可扩展的软件平台来实现软硬件接口。

目前国内外已有部分相关的软件开发平台，但存在各种各样的局限性，有的只能满足某个领域测控软件的开发需要；有的可扩展性差，无法满足惯导系统的测控需求。因此，需要针对惯导系统的测控要求，有针对性地开发适用的测控软件开发平台，切实提高测控软件开发效率和质量。

## 2 惯导系统测试需求

在对惯导系统进行测试时，具有以下特点和需求：

- 1) 通信接口类型多，包括RS422/232/485、

CAN 总线、1553B 总线、ARINC429 等；

2) 通信协议种类多, 不同的通信接口, 其通信协议通常不同, 即便是同一种通信接口, 其通信协议也有很多种类型;

3) 对测控软件运行实时性要求高, 测控软件要能够实时发送 (转发) 控制指令, 接收解析、显示并保存惯导系统发送的数据;

4) 在转台上进行标定、验证试验时, 标定方法种类多, 数据保存格式及数据处理方法各异;

5) 在进行导航测试时, 不同惯导产品的导航测试流程差异很大, 对导航数据处理的方法各异。

惯导系统存在的上述测试需求, 要求软件平台的体系结构必须是通用的、开放的和标准化的, 便于对其进行升级和扩展; 同时要具有很好的灵活性, 其中的组件要可重用和可移植, 以便实现软件平台中资源的共享。

### 3 软件平台体系结构设计

根据惯导系统测试需求, 由测控软件实现的

需求可以按照功能进行分类, 例如: 用于底层通信的硬件驱动接口模块、数据解帧组帧模块、数据处理模块、显示模块、转台控制模块等, 上述独立功能模块适宜采用插件方式进行设计、开发。

采用插件方式实现的功能可以有效地实现对外接口的统一, 同时能够很好地满足可扩展性要求。设计出的软件平台体系结构如图 1 所示, 由两部分组成: 一部分为软件平台的主体, 定义为框架; 另一部分为功能实现和扩展, 定义为插件。软件平台框架和插件都遵循统一的接口规范, 之间可以相互通信。插件是开发满足特定功能的组件集合, 框架可以加载符合软件平台接口规范的插件, 用户可通过增删或修改插件来调整软件平台的功能, 因此框架具备良好的可扩展性。

将开发完成的插件载入软件平台中, 然后利用软件平台框架提供的组态开发功能, 灵活地将相关的功能插件连接组装在一起, 生成针对某个具体惯导系统的测试控制流程, 用于惯导系统的测试。

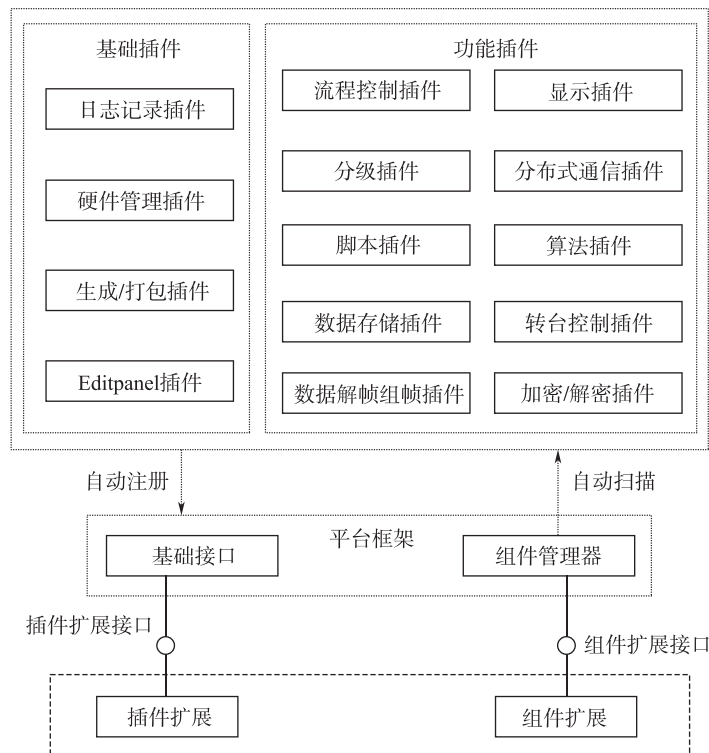


图 1 软件平台体系结构

Fig. 1 Software platform architecture

### 3.1 软件平台框架的功能

软件平台框架是一个通用的可独立运行的程序，定义了软件平台的接口规范，他可自动加载符合软件平台接口规范的插件，是整个软件平台的基础和主干，主要完成以下功能：

1) 负责启动整个软件平台和注册插件：软件平台启动后初始化窗体界面和组件管理器，扫描插件目录，完成插件的注册，并保存插件信息。

2) 负责管理组件和插件：完成组件、插件接口的初始化，加载和卸载组件、插件，并处理软件平台异常。

3) 负责向外界提供使用插件功能的界面（如菜单、工具栏等），通过规范接口调用工厂类组件实例化功能组件。工厂类组件是框架实例化功能组件的入口，为框架提供功能组件的相关信息，并根据需要实例化相应的功能组件。

软件平台中的插件包括基础插件和功能插件。基础插件在软件平台启动时由框架自动加载，仅实例化一次，初始化软件的界面和基本功能；功能插件辅助软件平台实现开发测控软件流程所需的具体功能。

### 3.2 组件及其接口设计

软件平台中以 XML 自描述的形式建立标准化的组件模型，并设计统一的对外接口。设计的组件基本接口如图 2 所示，包括 3 个基本接口：Report ()、Notify () 和 Attach ()。其中，Attach () 接口用于实现组件间的注册，Report () 接口与 Notify () 接口共同实现组件间的信息交互，组

件间可采用数据交换和消息传递两种方式进行通信。组件间的数据交换采用端口到端口的传输机制，若传输的数据量较大可采用缓冲区机制；组件间进行消息传递时，采用发布-订购机制传输 XML 格式的消息。

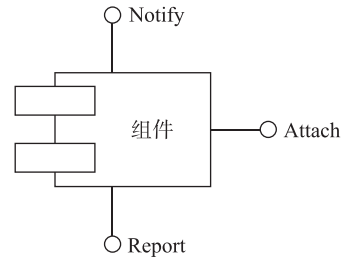


图 2 组件接口

Fig. 2 Component interface

软件平台内的所有组件均继承上述 3 个基本接口，如图 3 所示。不同类型的组件根据其实现的功能对外提供不同的应用接口，有以下三类典型的组件：

1) 工厂类组件：提供了 CreateComponent () 接口，该接口可用于实例化功能组件。

2) 图示类组件：主要负责实现测控软件界面设计所需的功能，不包括测控软件所需的计算或处理功能；提供了 Draw () 接口，该接口可根据需要在流程组态视图和界面组态视图内绘制相应的图形，若组件是非可视化的，则 Draw () 接口为空。

3) 功能组件：按照设计要求对外提供 Run ()、Draw () 和 PopPropertyDlg () 接口，分别实现运行、绘制图形和属性设置功能。

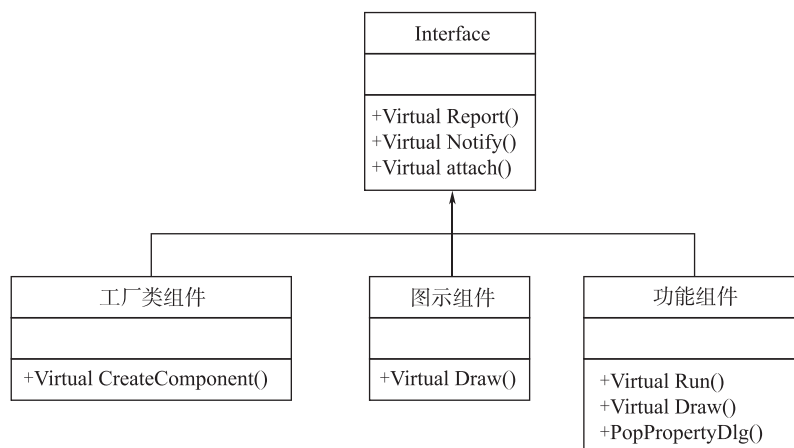


图 3 组件接口继承关系

Fig. 3 Inherit relationship of the component interface

### 3.2 插件设计及开发

插件是一系列相关功能的集合体，被封装在同一个动态链接库中，包含一个或多个功能组件和一个工厂类组件，软件平台中以插件作为组件的组织形式。软件平台启动时自动扫描插件目录，并将可用插件注册到软件平台中，插件的工厂类组件提供的接口供软件平台调用，同时也是功能组件实例化的入口。采用插件技术可实现软件平台的功能定制，提高软件平台的灵活性。

#### 3.2.1 插件基本结构

插件由“工厂类组件”和“功能组件”两部分组成，工厂类组件是一组与软件平台相衔接的接口函数，功能组件负责实现特定的功能。插件的基本结构如图 4 所示，这里的“工厂类组件”与图 3 中的定义相同，除了组件所具有的 Report ()、Notify () 和 Attach () 等基本接口外，还提供了 CreateComponent () 接口，用于实例化插件内的功能组件。

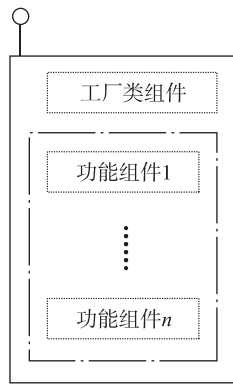


图 4 插件基本结构

Fig. 4 Plug-in basal architecture

#### 3.2.2 插件与软件平台框架的关系

软件平台以动态链接库的方式实现插件，插件与软件平台框架的关系如图 5 所示。插件中包含的信息如插件名、插件动态库路径、插件图标路

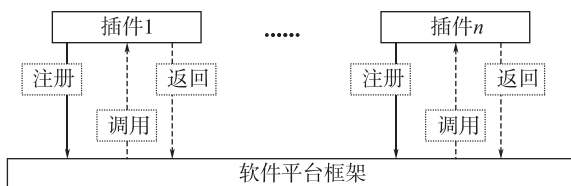


图 5 插件与软件平台框架的关系

Fig. 5 The relationship between plug-in and software development platform

径等，均以 XML 格式进行描述。软件平台通过读取 XML 文件获取相应信息并注册插件，插件注册成功后，软件平台就可以调用插件中的功能组件完成特定的任务，以此实现测控软件开发过程中所需功能及软件资源的“即插即用”。

#### 3.3.3 核心插件 Editpanel

##### 3.3.3.1 Editpanel 插件的功能

Editpanel 插件是整个软件平台的核心部分，用于实现拖放式组态、组件通信及工程的创建、打开和保存。Editpanel 插件负责对用户操作引起的事件做出响应，实现用户与软件平台的交互。基于状态机思想设计的 Editpanel 插件实现了双视图结构（即界面视图和组态视图），并为用户提供了三种工作模式：设计模式、运行模式和调试模式。

设计模式下，可以在界面视图和组态视图之间自由切换，界面视图提供各种界面图标，用于设计用户界面；组态视图提供具有端口的图标，通过拖放式组态完成测控软件控制流程的设计。设计完成后，切换至运行模式，可在界面视图内观看界面运行效果、查看测试数据；如果切换至调试模式，则可在组件端口处设置断点，通过单步执行、执行至断点等方法清晰地观察数据的流动。测控软件运行时，通过拓扑排序算法确定组件间的时序拓扑关系，保证每个组件接收的数据都是上一拓扑节拍产生的最新数据，满足数据传递的实时性及时序关系的正确。

##### 3.3.3.2 Editpanel 的状态机模型

Editpanel 插件基于状态机思想进行设计，用户的操作作为状态转移的触发条件，采用表驱动的方式建立有限状态机，对应的状态转移表中保存了当前状态、触发事件、下一状态以及相应的处理函数。其中触发事件与处理函数用宏表示，状态转移表格式如表 1 所示。

表 1 状态转移表格式

Tab. 1 The format of state transfer table

当前状态	触发事件	下一状态	执行动作
------	------	------	------

状态机模型如图 6 所示，在状态机运行过程中，当有事件触发时，状态处理函数 StatusHandle () 查询状态转移表，查询成功后转移至新的状态并进入事件处理函数 ActionHandle ()。



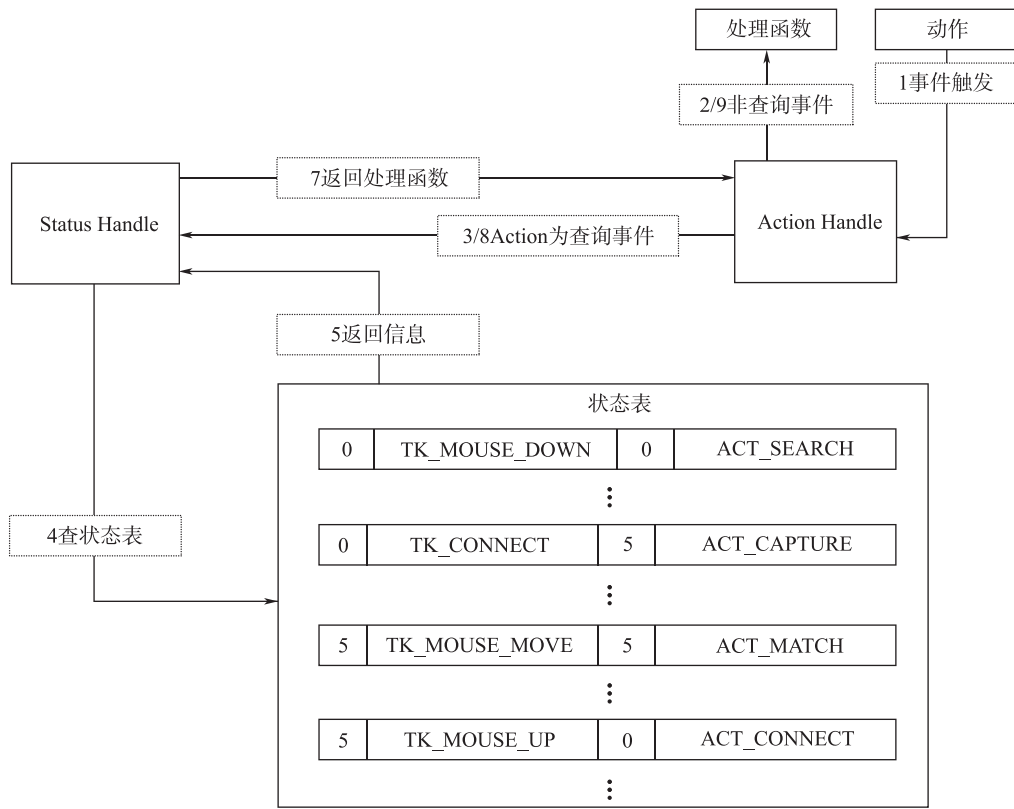


图 6 状态机模型

Fig. 6 State machine model

为保证软件平台的双视图结构正常工作，设计了 DisplayEditpanel 容器和 LinkViewEditpanel 容器，两个容器互为兄弟节点。DisplayEditpanel 容器负责用户与界面视图的交互，LinkViewEditpanel 容器负责用户与组态视图的交互。当用户与其中一

个容器交互时，该容器向兄弟节点发消息，完成双视图的同步更新。

DisplayEditpanel 的状态迁移如图 7 所示，LinkViewEditpanel 的状态迁移如图 8 所示。

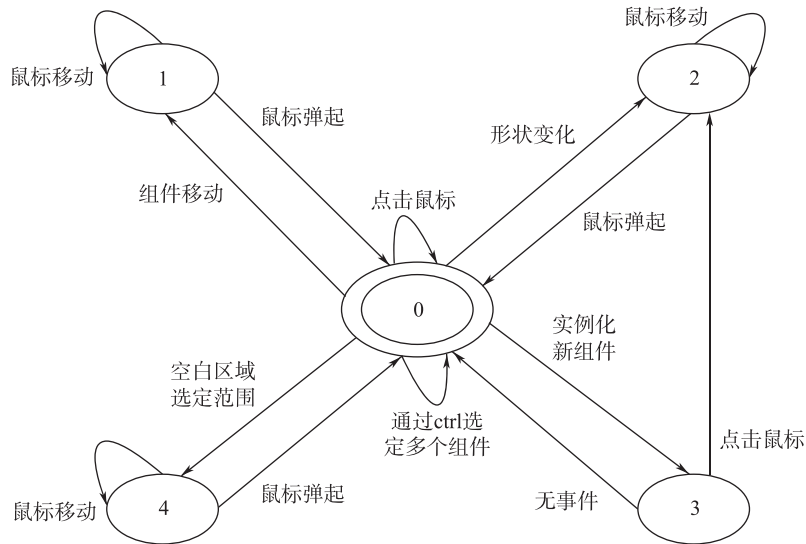


图 7 Display Editpanel 的状态迁移图

Fig. 7 Display Editpanel state transfer chart

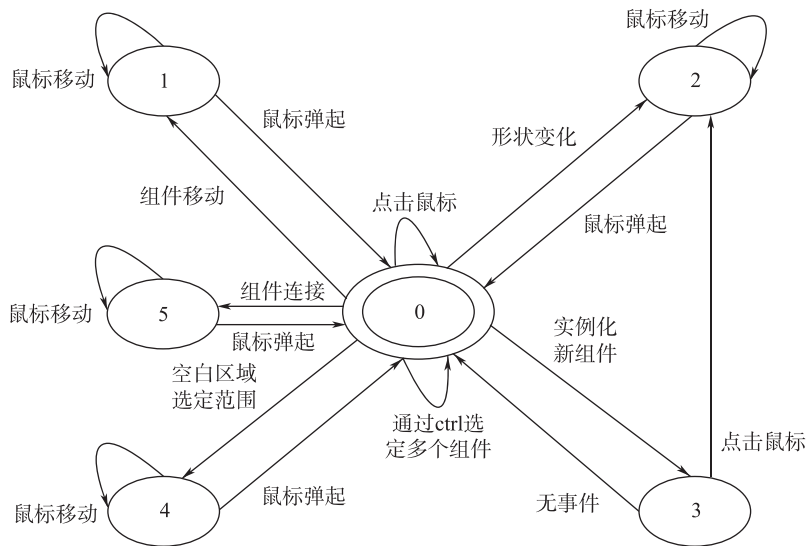


图 8 LinkViewEditpanel 的状态迁移图  
 Fig. 8 LinkViewEditpanel state transfer chart

软件平台启动后，通过状态机完成设计模式下软件平台与用户的交互。状态机默认以零号状态开始，等待用户的操作触发相应事件；状态机根据触发事件调用事件处理函数完成事件的响应，并调用状态处理函数完成状态的迁移。

### 3.4 软件平台的可扩展性

可扩展性是该软件平台最大的优势和特色，由于软件平台的功能是通过各种插件实现的，只需封装符合软件平台接口规范的新插件，将其放在软件平台的扫描路径下，软件平台运行时就可以自动加载新开发的插件，实现软件平台功能的扩展。

为了满足惯导系统测控要求，软件平台中内置了一些定制组件，如串口通信卡组件、CAN 通信卡组件、1553B 通信卡组件、惯导自检组件、惯导封装组件、惯导对准组件、惯导导航组件、转台控制组件、数据处理组件等，方便了惯导用测控软件的设计开发。

该软件平台的应用方式虽然与传统的组态软件类似，但工作原理有很大区别，软件平台运行时直接执行各种插件、组件的 .dll 文件，因此能够满足测试时对实时性的要求。

## 4 测控软件开发模式

软件平台运行后的主界面如图 9 所示，界面上主要包括 3 个区域，左上部为“项目管理”区，

右上部为“界面组态”区，下部为“流程组态”区。在利用软件平台开发测控软件时，以可视化组态的方式进行设计开发，具有固定的步骤和流程。

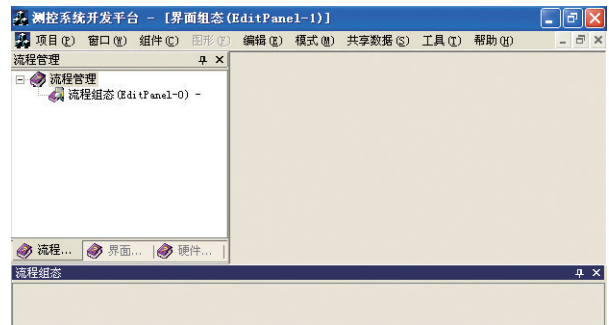


图 9 测控软件开发平台主界面  
 Fig. 9 Main interface of the software development platform

惯导系统测控软件基本工作流程如下：

- 1) 测控软件接收惯导系统自检数据帧；
- 2) 自检正常后进行惯导系统参数装定；
- 3) 装定正确后发送对准指令、转导航指令；
- 4) 测控软件接收惯导系统导航数据。

下面根据惯导系统测控软件工作流程，以一个实例描述测控软件设计开发过程。该实例通过“开始测试”按钮控制测试过程的启动和停止，利用“数据装定”、“对准指令”、“导航指令”等按钮进行指令发送，设计完成的测控软件如图 10 所

示。软件平台运行后默认为编辑模式，在该模式下进行测控软件的设计，具体步骤如下：

1) 选择“项目”中的“新建”菜单，新建一个工程。

2) 调出所需的工具栏，如基础控件类、显示类、存储类等，工具栏中的每一个可视化图标都是一个封装好的组件。

3) 根据需要将“流程组件”图标和所需的功能组件图标拖放到界面下部的“流程组态”区域，如果“流程组件”具有界面特性，软件平台将自动在“界面组态”区域放置相应的“界面组件”图标。用到的组件包括：按钮组件、RS422板卡组件、帧解析组件、数据列表组件、数据存储组件等。

4) 根据需要调整“界面组件”的大小，并利用软件平台提供的组件排布工具将各组件放置在合适的位置。

5) 在组件之间进行连线，有两种方式：一种是直接用线条连接不同组件的端口；另一种是无线连接，通过不同的端口颜色确定连接关系。完成连线后测控软件设计过程结束。

测控软件设计完成后，在运行模式下点击“开始测试”按钮，测控软件开始接收、解析、显示并保存惯导系统数据，测试人员根据需要向惯导系统发送控制命令，实现对惯导系统的测试。

行的可执行程序，测控软件整个开发过程结束，否则重新返回到“编辑模式”修改设计。通过上述设计开发过程可以看出，无需编写源代码，使得测控软件的开发效率大幅提高；由于测控软件是软件平台中若干组件的集合，通过修改组件属性及组态流程，便可适应对不同惯导产品的测试，因此软件平台具有很强的通用性。

### 5 结论

本文对测控软件通用开发平台进行了研究，在对惯导系统测试需求进行分析的基础上，设计了软件平台体系结构，详细论述了软件平台框架的功能、组件和插件的设计、状态机模型及平台的可扩展性，最后通过一个简单的实例，描述了利用软件平台开发测控软件的开发模式。利用软件平台开发惯导系统测控软件时无需编写源代码，测控软件是由一系列标准的插件组合而成的。利用该软件平台开发惯导系统测试软件，不但可以有效地提高开发效率和产品质量、降低人力成本，而且使测控软件的标准化的系列化、模块化程度大幅提高。

#### 参考文献

- [1] 刘畅, 李显甫. 一种自动测试系统的开发平台-PAWS[J]. 测控技术, 2003, 22 (12) .
- [2] 王红, 杨小辉. PAWS 软件平台的资源描述及自动配置实现分析[J]. 测控技术, 2004, 23 (9) .
- [3] 王红, 王湘念. 自动测试系统软件平台发展现状及建议[J]. 航空制造技术, 2008, 22.
- [4] 于功敬, 马好东, 杨广志, 奚全生. 虚拟仪器测试环境软总线体系结构[J]. 计算机测量与控制, 2006, 14 (2) .

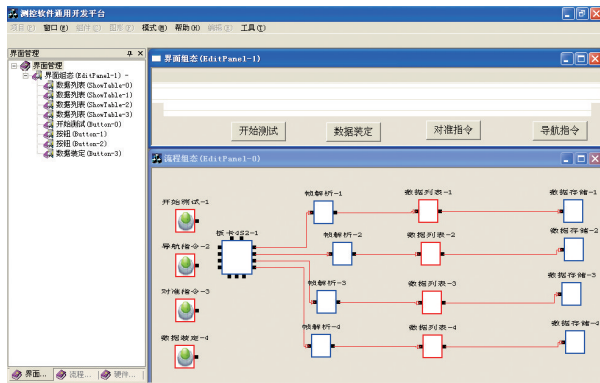


图 10 设计完成的测控软件

Fig. 10 The designed test software

利用软件平台设计的测控软件，运行后如果满足设计要求则可通过软件平台提供的“发布”功能，将测控软件发布为可脱离软件平台独立运