

基于 GJB6600 的 IETM 逻辑引擎设计

乔福超

(海军航空工程学院, 山东 烟台 264001)

摘要:为提高国产交互式电子技术手册(IETM)的人机交互能力,论述了 GJB6600 中过程数据模块的结构和作用,分析了 IETM 逻辑引擎的运行机理,提出了逻辑引擎的 2 种实现方法。使用 JavaScript 语言设计并实现了基于 GJB6600 的编译型逻辑引擎。对 IETM 逻辑引擎的设计与开发具有一定的参考价值。

关键词:交互式电子技术手册;逻辑引擎;GJB6600

本文引用格式:乔福超. 基于 GJB6600 的 IETM 逻辑引擎设计[J]. 四川兵工学报, 2015(4): 116 - 119.

Citation format: QIAO Fu-chao. Design of IETM Logic Engine Based on GJB6600[J]. Journal of Sichuan Ordnance, 2015(4): 116 - 119.

中图分类号: TP391.1

文献标识码: A

文章编号: 1006 - 0707(2015)04 - 0116 - 04

Design of IETM Logic Engine Based on GJB6600

QIAO Fu-chao

(Naval Aviation Engineering Institute, Yantai 264001, China)

Abstract: In order to improve the level of human-computer interaction in domestic interactive electronic technical manual (IETM), the process data module's structure and function were discussed. The IETM logic engine's mechanism was also analyzed. Two methods of realizing logic engine were proposed. A kind of logic engine based on GJB6600 was designed, which was edited by JavaScript. It has certain reference value for the design and development of IETM logic engine.

Key words: interactive electronic technical manual; logic engine; GJB6600

交互式电子技术手册(interactive electronic technical manual, IETM)是以数字形式存储,采用文字、图形、表格、音频和视频等形式,以人机交互方式提供装备基本原理、操作使用和维修内容的技术出版物^[1]。与传统的电子技术手册相比,IETM 最显著的特征是具有交互性,这使得用户与计算机之间能够以人机对话的方式获取信息与知识,极大地提高了装备维修效率。

IETM 普遍采用过程数据模块(Data Module)和逻辑引擎(Logic Engine)相互配合的方式来实现人机交互功能。目前,针对欧洲 S1000D 标准的 IETM 开发平台已经具有了成熟的逻辑引擎。我国自 2009 年颁布 GJB6600 标准以来,也出现了各型支持该标准的 IETM 开发平台,但由于平台的侧重点不同,目前所接触的几款支持 GJB6600 标准的开发平台都暂不提供对过程数据模块和逻辑引擎的支持。因此,设计一种基于 GJB6600 标准的逻辑引擎,为相关的研究与开发提

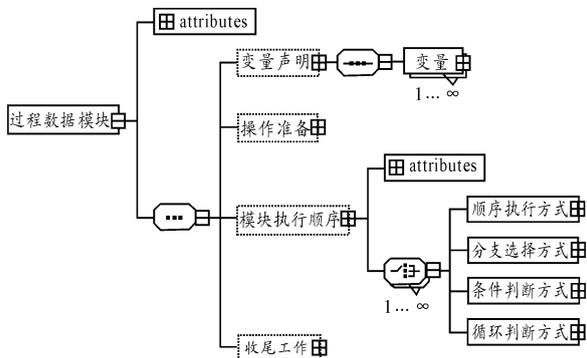
供参考。

1 过程数据模块

IETM 中的技术信息是以数据模块的形式进行存储的。数据模块是描述装备结构、性能和操作步骤等技术信息的独立的、完整的数据单元。根据技术信息的用途不同,GJB6600 将数据模块分为 8 类,包括描述类、程序类、故障类、维修计划类、操作类、图解零件类、连线类、过程类。其中,过程数据模块是一类特殊的数据模块,它是具有内在逻辑判断的脚本流,能够根据装备的状态信息组织其他数据模块和步骤的显示顺序。

GJB6600 所定义的过程数据模块包括〈变量声明〉、〈操作准备〉、〈模块执行顺序〉和〈收尾工作〉子元素,其数据模型如图 1 所示。其中,〈变量声明〉和〈模块执行顺序〉是实

现 IETM 交互性的 2 个重要元素。下面对过程数据模块中的主要元素作简要分析。



1.1 〈变量声明〉元素

在装备维修过程中, IETM 经常需要获取一些与装备相关的状态信息(如温度、输出电压等),以便判断故障类型,指引用户维修。〈变量声明〉元素即是用于对维修过程中所需的所有状态信息(变量)进行定义和初始化的。在 IETM 运行过程中,使用逻辑引擎驱动对话框或调用外部测试程序,以使用户能够手动输入信息或由自动测试设备自行输入信息。逻辑引擎将所获取的状态信息以变量的形式存储于状态表中,以供后续的逻辑判断、数值计算或其他程序使用。

1.2 〈模块执行顺序〉元素

〈模块执行顺序〉元素用于描述多个数据模块和步骤之间执行的先后顺序,形成具有内在判断逻辑的脚本流。它包括〈顺序执行方式〉、〈分支选择方式〉、〈条件判断方式〉、〈循环判断方式〉4 种子元素。其中,〈顺序执行方式〉是核心元素,其作用是描述向用户呈现的信息内容,包括独立的操作步骤、对话框、消息框、引用的数据模块、外部请求等内容;〈分支选择方式〉描述多分支情况下的业务逻辑,逻辑引擎通过判断其子元素的〈适用性信息〉元素值或〈适用性引用〉属性值,决定在特定的逻辑条件下所应选择的分支;〈条件判断方式〉用于描述一个 if - then - else 的业务逻辑,逻辑引擎通过判断其〈表达式〉元素值决定应选择的分支;〈循环判断方式〉描述一个循环逻辑,逻辑引擎通过判断其〈表达式〉元素值决定是否跳出循环。需要指出的是,上述 4 种元素均可在其子元素中继续套用这 4 种结构,以描述任意复杂程度的脚本流。

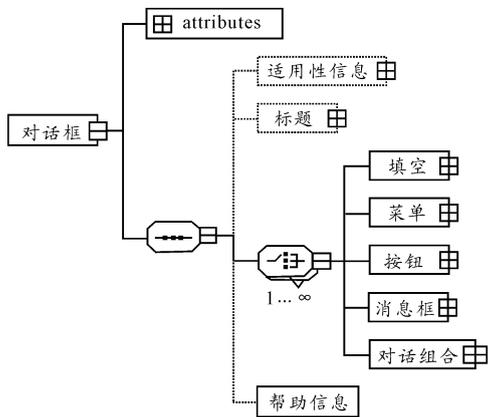
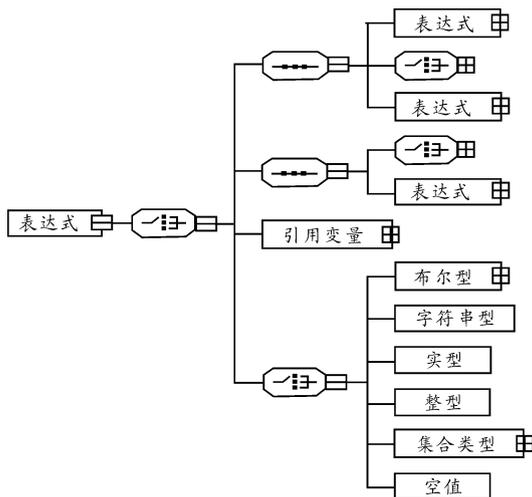
1.3 〈表达式〉元素

〈表达式〉是过程数据模块中极为重要的元素,在脚本流中,所有的逻辑表达式、算术表达式、直接量等都需要使用〈表达式〉进行描述。该元素具体包含 4 种结构的子元素,如图 2 所示。

1.4 〈对话框〉元素

人机对话是 IETM 交互功能的基础。在装备维修过程中,当逻辑引擎需要获取或更新某一状态信息(变量)时,会弹出对话框并要求用户输入相应的信息。〈对话框〉元素即

用于描述所弹出对话框的内容以及与之相关联的变量。该元素主要包含 5 种类型的子元素,即〈填空〉、〈菜单〉、〈按钮〉、〈消息框〉、〈对话框组合〉子元素,通过搭配使用这 5 种子元素可实现简洁高效的对话框。〈对话框〉元素的数据模型如图 3 所示。



2 逻辑引擎

2.1 逻辑引擎运行机理

过程数据模块是静态的脚本,其本身无法实现人机交互。逻辑引擎是解析执行静态脚本的一个软件,是脚本流的解释器^[2]。在 IETM 中,过程数据模块必须借助逻辑引擎的支持才能完成人机交互功能。

本节以一段维修流程为例,对过程数据模块和逻辑引擎的作用进行说明,如图 4 所示。首先,数据模块编制人员将由厂商提供的设备维修流程按照一定的 IETM 标准编制成过程数据模块(该项工作通常是借助 IETM 开发平台完成的,优秀的开发平台能够使用图形界面来构建维修流程,并自动生成基于 IETM 标准的过程数据模块)。在数据模块编制完

成后,逻辑引擎对过程数据模块中的静态脚本进行解析并处理。逻辑引擎的运行涉及2方面内容:解析器和处理器。其中,解析器负责对XML文档中的节点进行遍历,并根据节点

的类型调用相应的处理器;处理器则根据当前所遍历的节点类型及内容执行相应的处理,如显示数据模块内容、弹出对话框等。

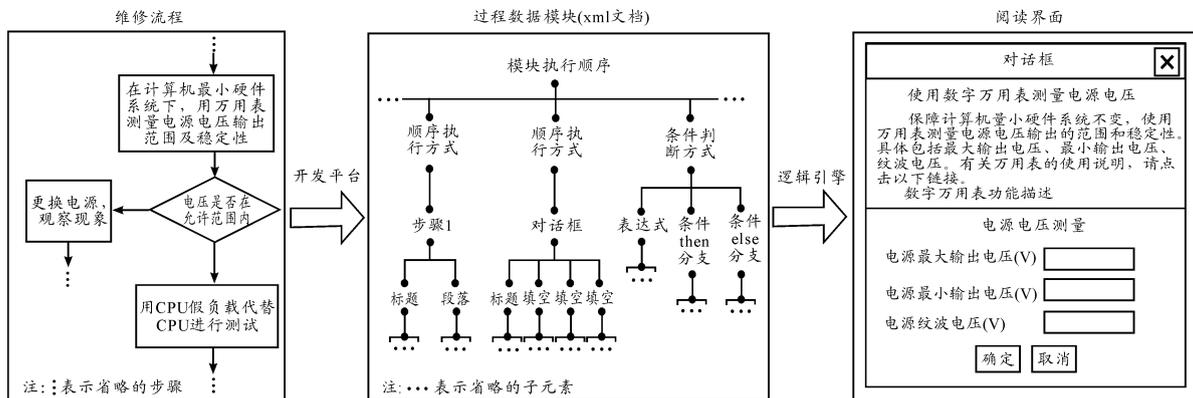


图4 过程数据模块和逻辑引擎的作用示意图

2.2 逻辑引擎实现方案

解析器与处理器有多种实现方式,本研究提出2种可行的方案。

1) 解释型逻辑引擎

该型逻辑引擎的运行机理类似于程序设计语言中解释型语言的翻译过程。它首先加载过程数据模块,然后由解析器逐个节点地进行遍历,每当遍历到一个节点,便调用处理器执行相应的操作。解析器会根据用户的操作和文档内在的逻辑有选择性地遍历XML文档,是一种动态的翻译方式。

2) 编译型逻辑引擎

该型逻辑引擎的运行机理类似于程序设计语言中编译型语言的翻译过程。它是将整个过程数据模块直接翻译成完整的可执行的程序代码。它首先加载过程数据模块,并对文档中的节点进行遍历。每当遍历到一个节点,即调用相应的翻译函数,将该节点的内容翻译成可执行代码,并以字符串的形式存储该代码。在遍历完成后,将所得到的所有节点的代码拼接形成完整的程序代码,并调用执行。

3 逻辑引擎设计

本研究使用JavaScript语言设计了一款基于GJB6600的编译型逻辑引擎。下面对引擎实现的关键环节和程序流程设计进行详细说明。

3.1 逻辑引擎关键环节

3.1.1 XML解析

XML文档的解析通常需要借助XML解析器来完成,本文即采用了基于DOM规范的解析器。DOM规范为解析XML文档定义了一组标准接口。DOM解析器首先读入整个文档,然后将该文档转换成常驻内存的树状结构,即DOM树。程序代码可以通过节点之间的关系、标签名、属性等访问DOM树中的节点,并获取每个节点所包含的数据。

3.1.2 翻译函数

编译型逻辑引擎的处理器的主要工作是节点内容翻

译成程序代码。因此,应依据GJB6600过程数据模块的数据模型,为每一种类型的元素设计相应的翻译函数。翻译函数将所要翻译的节点作为参数传入,将节点内容翻译成程序代码,并存入一个字符串变量作为函数返回值。以〈表达式〉节点为例,其对应的翻译函数代码如下所示:

```
function dealExpress(Node)
```

```
{
    var str = [];
    var childs = Node.childNodes;
    var j = 0;
    for(j = 0; j < childs.length; j++)
    {
        switch(childs[j].nodeName)
        {
            case "表达式":
                str = str + "(" + dealExpress(childs[j]) + ")";
                break; // 递归
            case "引用变量":
                str = str + childs[j].getAttribute("变量名称");
                break;
            case "小子": str = str + "<"; break;
            case "大于": str = str + ">"; break;
            case "整型":
                str = str + childs[j].childNodes[0].nodeValue;
                break;
            case "实型":
                str = str + childs[j].childNodes[0].nodeValue;
                break;
            case "与": str = str + "&&"; break;
            ... // 以下省略
        }
    }
}
```

```
return str;//返回包含代码的字符串
```

图 5 所示的是一个表达式节点的 XML 树形图,该节点描述了一个逻辑表达式,值为: $(V_Min > 11) \&\& (V_Max < 13) \&\& (V_Ripple < 0.5)$ 。使用上述翻译函数对其进行翻译,将其最顶层表达式节点作为参数输入该函数,所返回的字符串值为: $(((V_Min) > (11)) \&\& ((V_Max) < (13))) \&\& ((V_Ripple) < (0.5))$,即完成了对该节点内容的翻译。

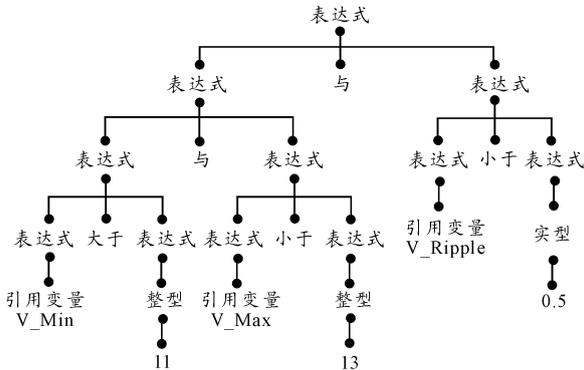


图 5 表达式节点树形结构

3.1.3 遍历顺序

编译型逻辑引擎最终将所有节点的翻译代码拼接形成完整的程序代码。代码的拼接顺序受节点的遍历顺序影响,因此必须选择合适的遍历方式。采用类似二叉树中先序遍历的方式,按自顶向下、自左向右的顺序,先遍历节点本身及其子节点,再遍历兄弟节点。

3.1.4 无限嵌套

在过程数据模块中,经常出现元素之间相互嵌套或元素嵌套其本身的情况,如图 5 中的 <表达式> 元素。在 XML 文档中,这种类型的节点可以具有任意多层次的子节点,其结构无法预测。因此,对这种可无限嵌套的元素设计翻译函数时,可使用递归调用的方法,如上述 <表达式> 元素的翻译函数所示。

3.2 逻辑引擎程序设计

基于 GJB6600 的编译型逻辑引擎的程序流程图如图 6 所示。该程序首先加载 XML 文档,并将其转化为 DOM 树,声明全局字符串变量 Str 以存储整个模块翻译完成的代码。然后对文档节点进行遍历,每当遍历到一个节点,即调用相应的翻译函数,将该节点的内容翻译成可执行代码,并以字符串的形式存储。当文档全部遍历完成后,Str 变量值即为完整的程序代码,调用 eval() 函数即可执行字符串中的代码。

4 结束语

论述了 GJB6600 标准中过程数据模块的结构和作用,分析了 IETM 逻辑引擎的运行机理,提出了逻辑引擎的 2 种实现方案,即解释型逻辑引擎和编译型逻辑引擎。使用 JavaS-

cript 语言设计实现了基于 GJB6600 标准的编译型逻辑引擎。研究成果对 IETM 逻辑引擎的设计与开发具有一定的参考价值。

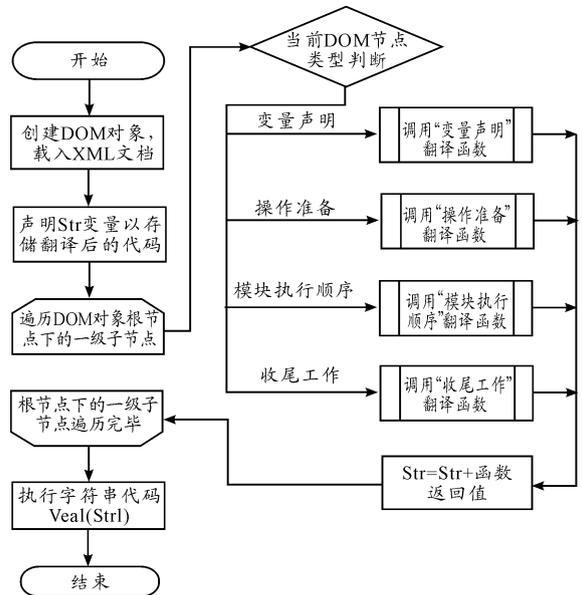


图 6 逻辑引擎程序流程

参考文献:

- [1] GJB6600, 装备交互式电子技术手册[S].
- [2] 寇建华, 左洪福, 苏艳. 交互式电子技术手册解析关键技术研究[J]. 电脑知识与技术, 2009, 5(28): 7941 - 7942.
- [3] 刘双双, 叶晓慧, 王红霞. 嵌入故障诊断策略的 IETM 的研究与实现[J]. 计算机测量与控制, 2010, 18(9): 1964 - 1966.
- [4] 朱兴动. 武器装备交互式电子技术手册—IETM [M]. 北京: 国防工业出版社, 2009.
- [5] 张丽丽, 畅峰. 基于 S1000D 的空空导弹系统 IETM 的制作[J]. 四川兵工学报, 2014, 35(5): 58 - 61.
- [6] 徐宗昌, 雷育生. 装备 IETM 研制工程总论[M]. 北京: 国防工业出版社. 2012. 1 - 44.
- [7] 徐宗昌, 孙寒冰, 朱伟成, 等. 基于 IETM 的交互式故障诊断模式研究[J]. 计算机测量与控制, 2013, 21(10): 2651 - 2654.
- [8] 于大海, 刘浩. 基于故障模式的交互式电子技术手册的设计与实现[J]. 四川兵工学报, 2010, 31(6): 28 - 30.
- [9] 崔建锋. 交互式电子技术手册—IETM 综述[J]. 航空制造技术, 2011(12): 105 - 107.
- [10] 刘松风, 林志文, 杨士元. 基于 IETM 驱动的智能便携故障诊断设备[J]. 计算机测量与控制, 2009, 17(6): 1030 - 1032.
- [11] ASD S1000D, International Specification for Technical Publications Utilizing A Common Source DataBase, Issue 4. 0 [S].