基于对象组件的遥控软件设计

欧余军 詹 磊
（北京航天飞行控制中心）

摘 要 通过分析飞控中心遥控软件系统存在的现状，提出基于对象组件的遥控软件设计思想，以及具体的设计步骤，并针对遥控软件给出类的抽象与设计，有助于遥控软件的开发与应用。

关键词 遥控软件 对象组件 设计

1 引 言

随着我国航天事业的不断发展，北京航天飞行控制中心（以下简称北京中心）承担的航天测控任务也越来越繁重。遥控软件作为中心测控软件系统的一个重要组成部分，在设计时由于受到当时计算机软硬件发展水平的制约，存在着系统耦合度高、扩展性差、开发效率低下、调试复杂等问题。

2 基于对象组件的体系结构

在基于对象组件的软件体系结构中，组件是可插入到语言、工具、操作系统、网络系统中的二进制代码和数据。一个组件由一组对象构成，它们共同工作来提供一种系统功能。定义一个基于对象组件的软件系统的结构框架和构成，最终只表现为一系列类（子类）、它们的定义和对象。对象控制整个系统并向其它对象发送消息，用户可以通过准确定义方法的集合向该对象发送消息来处理这个对象。

基于对象组件的软件体系结构为设计大型应用软件系统给予了强有力的支持，它是大型系统中又一种具有某种功能的封装体，除了它本身以外，且声明和对象对系统的其它部分是不透明的。这意味着它不可能和系统的其它部分有相互依赖的关系，它封装的所有对象都被隐藏起来了。这样组件的作用就好像是一个软件意义上的集成电路，可以通过一些组件按照一定的逻辑关系来构建软件系统。

利用基于对象组件的体系结构就可以更好的处理遥控软件内部的复杂度和依赖性，加强软件的聚合性、独立性和重用性。而在遥控软件内部的某一个功能模块中，软件开发者则可以充分运用面向对象的体系结构，通过对象的封装性、多态性和继承性，降低对象间的耦合度，提高对象的可扩展性。

3 当前遥控软件的不足

当前遥控软件的体系模式是在中心测控软件系统开发初期确定并设计完成的，参加过多次载人航天飞行测控任务，前后使用了近 10 年时间，由于受到当时各种条件的制约，存在一些不足之处。

3.1 软件可重用性差

在遥控软件设计初期，首先根据软件需求将功能模块自顶向下抽象分解，然后各个模块由不同的开发人员实现。尽管底层公共服务库为各个功能模块提供了一定的支持，但由于没有良好的刚性机制保证，而且公共服务的粒度过小，导致软件代码的重用性不高。

3.2 软件耦合度高

由于受到实现方法上的限制，软件各个实现模块没有实现封装，特别是软件底层的公共服务，在多个模块中都有调用。对于这些服务的修改，将会导致上层遥控软件中多处都要做相应的修改。

3.3 公共库缺乏有效的组织管理

虽然建立了遥控软件公共服务库，但是公共库函数的使用说明不清晰、缺少维护和发布机制。开发人员无法方便地了解有什么样的公共库可以使用，使得公共库的使用率不高。

4 基于对象组件的遥控软件

遥控软件的设计，应继承原有软件封装系统服务和初始化信息等优点，继续提高软件内聚性并降低耦合性，同时仍然保持程序与数据分离以及多平台支持的特点。

4.1 设计思想

新的遥控软件设计采用了基于对象组件的的设计思想,它把问题域作为一系列相互作用的对象来构造模型,对象的软件模型与模型间的关系汇集成软件系统的基本结构，它以一种与人类思维方式相似,但更直接、更自然的方法来描述和处理遥控中的问题,用这种思想设计和开发软件,可以在很大程度上提高软件的可扩充性、可维护性、开放性和分布性。

基于对象组件的遥控软件体系结构是建立在一系列的类上,这些类刻画了遥控中所有要处理的基本数据类型的行为,而后将这些类组装成一些组件为系统提供服务。具体来说,在遥控软件中,一组对象的组合就可以很好的处理软件内部的复杂度和依赖性,加强软件的聚合性、独立性和重用性。在遥控软件内部的某一个功能模块中,软件开发者可以通过对象的封装性、多态性和继承性,降低对象间的耦合度,提高对象的扩展性。在这种基于对象组件的开发方式中,软件组件是实际被部署的单元,是软件系统中独立的部分,它可以是一个类,也可以是被编译并链接到包里的一组类。由于概念的封装和实现的隐蔽,使得类具有更大的独立性,这样我们就能在任一时刻在类的层面上增加新的操作,并能够修改实现,以改进性能,或引入原来设计中没有的新服务,这就为我们在对公共库服务的修改维护上提供了一个很好的机制。而逻辑上作为组件实例组成部分的对象,是根据已经部署到组件中的类在需要的时候被实例化的。所有这些都很好地弥补了当前遥控软件中的不足。

4.1.1 为业务处理层“减肥”

未来的开发者开发的将是只有在执行的时候才装订到一起的网络服务联合起来的应用软件。在基于对象组件的遥控软件的实现中，我们把焦点集中在类和类层次结构的设计、实现和重用上,提高了软件的开放性和可扩展性。

4.1.2 为底层服务“增肥”

现有的遥控软件提供了一些的底层公共服务，这些服务在一定程度上增加了系统的开放性。但由于受设计方法的限制,这些服务仍然显得数量较少、粒度较小。

在新软件中，对现有的公共服务进行了重新整理，并在这些基本的公共服务基础上建立了更高层次的组合服务，使这些组合服务可以完成某个业务流程中的一段功能，从而为更高层的业务处理层提供强有力的支持,大大增强了系统的灵活性。在这些公共服务的开发过程中，利用系统预编译等方法提供对多种操作系统平台如 VMS、WINDOWS、UNIX 的支持，使遥控软件的开发可以不受系统平台的限制,提高了系统的可移植性。

4.2 设计实现

遥控软件的设计以对象和类为中心,它将整个遥控系统分解成基本数据类或者子类,以及每个基于类和子类的特殊定义。遥控软件的静态关系如图 1 所示。

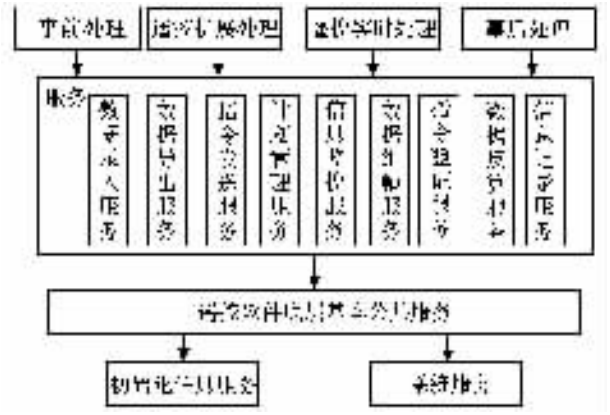

图 1　新遥控软件静态关系图

4.2.1 设计步骤

整个软件开发工作分成分析、设计和实现三个阶段，其中分析阶段的主要工作是识别遥控任务中的基本对象及其相互关系；设计阶段的主要工作是将这些对象及其相互关系模型化,并建立分类关系；实现阶段的主要工作是针对具体的应用问题将基本构件进行组装。

在对遥控任务分析以后的具体设计和实现阶段的步骤是：

(1)建立遥控软件的动态模型

· 根据遥控任务的具体要求确定组成遥控软件系统的对象及其应具备的固有处理能力。

· 分析对象间的联系，确定其相互间的消息传递方式。

· 根据上述两点设计对象的消息模式。

(2)建立遥控软件系统的静态模型

· 确定类,根据外部特性区分对象的类别。

· 建立类间的继承关系。

· 根据上述两点设计各对象的外部特性和层次结构。

(3)实现

· 类的开发,实现类的功能。

· 创建所需对象,实现这些对象之间的联系。

4.2.2 类的设计

遥控软件的设计首先考虑的是如何确定基本类,用它实现系统内部的数据通信和同步,构成一个满足用户功能性能要求的系统。我们使用消息类和共享内存类进行数据通信和同步，使用网络通信类和前端控制软件进行通信。消息类和共享内存类实际上是利用共享内存机制实现其方法集合的，网络通信类则利用 TCP/IP 协议实现其内部方法的。

遥控系统软件通过使用消息类和共享内存类进行系统内部进程间的通信,每个子系统可能使用类的不同对象,所以必须保证不同进程能识别同一消息队列和共享缓存区。共享内存类实现和消息类实现的机制是相同的,同类的不同对象通过构造函数实现指向内核中的同一共享内存区。网络通信类利用 TCP/IP 网络协议服务和套接字接口实现类中的方法。

根据遥控系统的设计要求和功能，我们可以将其划分为不同的目标对象，主要分为遥测数据的采集和处理部分,遥控流程的实现部分,接口部分和遥控显示部分。按需要分成几个类:遥测数据采集类、接口控制类、遥控界面控制类、遥控类、遥控服务类、遥控服务实现类，每个类中封装了用以完成各自功能的数据和函数，提供外部调用的数据和函数作为类的公有部分。其中公有函数的接口尽量简洁,以便于外界使用;私有函数的设计则注重模块化,用相对短小的程序段实现有限的功能,以便于程序的编写,调试和维护。类的层次关系图如图 2 所示。

图 2　遥控软件类的层次关系图

其中遥测数据采集和处理部分主要进行遥测数据的采集和处理，这些遥测数据包括遥控所需要的各种遥测参数；接口部分主要完成遥控系统与其它部分的接口，主要功能是完成接口数据的读取和数据发送等;遥控类是业务流程的模板,它的主要作用是遥控业务流程的实现；遥控服务类为遥控类提供遥控服务接口支持；遥控服务实现类具体实现遥控服务类中的接口函数。

4.2.3 系统组装

在确定系统的类以及将这些类组织成层次结构、并实现类的方法后,我们开始实现系统的功能。我们要建立对象的实例，这些对象对应于在分析阶段所标识的实体。最后我们建立实例之间的通信信道，这些信道可以通过把引用从一个对象传递到另一个对象来建立。

5 结束语

基于对象组件技术的应用，较好的改善了原有遥控软件中系统耦合度高、扩展性差、开发效率低下等问题,在一定程度上提高了遥控软件的开放性、扩展性、可移植性和软件效率。底层公共服务和遥控类的设计是遥控软件开发的基础，提供一套应用更广泛的公共服务,对遥控类的设计进行优化组合,将是我们今后研究工作的重点。质量和效率具有十分重要的意义。 ◇

参 考 文 献

[1] 龙守谧.面向对象与软件重用.小型微型计算机系统,1994.3

[2] 陈世鸿.面向对象软件设计方法的研究.武汉大学学报,1995.6

[3] 杨芙清.浅论软件技术发展.计算机应用文摘,1997.6

[4] 杜承烈.分布式测控系统面向对象的软件结构框架.西北工业大学学报,1999.2