

小卫星软件共用平台设计

廖明宏, 吴翔虎, 程光明

(哈尔滨工业大学计算机系, 哈尔滨 150001)

摘要: 以小卫星共用平台相配套, 以面向对象的程序设计方法 (OOP) 为基础, 提出软件共用平台的设计。利用 OOP 的封装性、继承性和多态性, 设计小卫星软件的基本类库, 介绍各个基本类的具体设计思想。最后给出由基本类库生成星载机软件系统的实例。实践表明, 软件共用平台对小卫星研制是行之有效的技术。

关键词: 小卫星; 软件平台; 面向对象程序设计; 星务软件; 姿态控制软件

1 引言

小卫星的开发和应用在航天领域成为一支新的生力军, 受到各国航天专家和学者的高度重视。它以“重量轻、体积小、价格低、周期短、性能高”的优势而倍受人们的青睐。尤其是中小型研究所或高校都在积极研制自己的小卫星, 如德国的 OHB 系统公司^[1]和英国的萨瑞大学^[2]。

一般称质量在 500~1000 公斤以下的卫星为小卫星^[3]。但质量绝不是唯一的指标。它采用先进的设计思想和技术路线来体现小卫星的综合优势。比如一体化的设计思想, 充分开发软件功能, 尽量减少硬件投入的思想和小卫星共用平台技术^[4]等在实验中被验证是行之有效的方法和技术。

小卫星共用平台可由 3 部分组成: 基本部分、推进部分 (包括冷或热气) 和太阳帆板与可展开的空间结构 (如天线等)。采用这种共用平台与不同有效载荷模块相结合可以组成不同用途的小卫星。小卫星平台的 3 部分可以独立存在, 根据不同要求来组合, 以满足不同要求。在进行不同组合时, 平台 3 部分技术尽量达到不更改, 或者基本上不更改。

与小卫星共用平台相类似, 本文提出软件共用平台的设计思想, 即采用面向对象的程序设计方法, 根据小卫星软件的不同功能, 建立基本类库。在设计不同的小卫星时, 可以由基本类库派生具体的对象来组合小卫星软件系统。采用软件共用平台的设计思想, 可以最大限度地重用那些经反复测试的、成熟的软件模块, 大大地缩短软件开发周期, 提高系统的可靠性。

探索一号 (TS-1) 小卫星是一种立体测绘科学试验卫星, 主要用于对地照相, 形成电子地图。小卫星上的计算机和各种控制器构成一个计算机网络系统, 该网络系统由两台 386EX 星载机及 4 个下位机 (通信控制器、电源控制器、GPS 控制器和有效载荷控制器) 通过两条 485 总线互联构成, 见图 1。其通信方式采用以星载机为主下位机为辅的主从工作方式。而两台星载机通过高速同步串口 (SSIO) 连接, 采用双机备份方式工作。每台星载机带有一个硬件看门狗, 当系统发生故障时, 产生中断, 必要时引起双机切换。

星载机的存储器包括: 16K 的 ROM: 存放系统引导程序; 256K 的 EEPROM: 存放应用程序; 1M 的 RAM: 用于运行程序; 72M 的固态盘: 用于存放遥测参数。

基金项目: 本文是国家高技术航天领域“863-2”基金资助项目 (编号: 863-2-5-1-8)。 **收稿日期:** 2000-07-26

作者简介: 廖明宏 (1966-), 男, 台湾人, 教授, 博士, 主要研究方向: 操作系统、人工智能和并行分布式处理。 吴翔虎 (1968-), 男, 黑龙江人, 副教授, 硕士, 主要研究方向: 操作系统。 程光明 (1971-), 男, 汉族, 黑龙江人, 博士生, 主要研究方向: 嵌入式系统。

姿态控制 I/O 部件包括:4 个飞轮和 3 个磁力矩器用于姿态控制;4 个陀螺、一个磁强计、两个星敏感器、一个 0-1 太阳敏感器、一个数字太阳敏感器用于确定卫星姿态。

本文以 TS-1 小卫星为应用背景,介绍小卫星软件共用平台的设计思想。

2 星上软件的基本类库

面向对象的程序设计方法是近几年来广为使用的程序设计方法。与传统的结构化程序设计不同,它根据事物的自然规律,提取其共性部分,将对数据的定义和对数据的操作

(这里称为方法)封装成对象,相同的对象可进一步抽象成“类”。由“类”可派生出具体的对象,对象之间通过消息传递方式相互作用。面向对象的程序设计方法为公共软件接口设计、软件的集成和重用提供了一个很好的框架。它的封装性、继承性和多态性为小卫星软件共用平台的设计提供了有力的支持。

封装性:是指“类”将对象的属性(即数据)和对象操作(即方法)封装成一个实体。这一原则有时也称为“信息隐藏”。小卫星上每一个部件可设计成一个独立的、封装的对象。其优点是每个对象只维护它自己的数据,执行自己的操作,并通过标准的接口返回其执行结果。其他对象只能通过指定的方法访问本对象的成员数据。从而使得外部接口标准化。封装的另一个好处是每一个对象只以参数的形式接收它们的数据,而不是以整个系统作为它们的信息,这对系统的调试和维护大有好处。

继承性:面向对象的程序设计方法提供一种从不同的“类”或对象中抽象出它们公共特性的机制,这一机制也称为“类”的继承性。它允许从已定义的“基类”中派生出“新类”,这些“类”的对象可以用“基类”定义的数据和操作。“类”的继承性在小卫星共用软件平台的设计中尤为重要。众所周知,卫星的研制是一个高可靠性的设计过程,人们极为重视那些被实践反复证明的、可靠的、成熟的器件或软件模块。从可靠的软件模块派生出新的对象在可靠性方面可得到一定的保证。小卫星共用软件平台的基本设计思想就是对所有的公共部分进行很好的抽象,而把那些特殊的属性留给具体的实例。对部件任何一个新的实现只需对那些涉及到具体的硬件或算法进行编码即可。

多态性:多态性原理允许来自不同“类”的对象对同一个消息做出反应,消息的接收者是在运行过程中动态确定的。这也称为“操作重载技术”。通过这种技术,允许那些具有相似属性的对象拥有一个公共接口。它根据被引用的对象的不同而执行不同的操作。

TS-1 小卫星星上软件包括星载机运行的软件和各个下位机运行的软件。由于星务管理软件和姿态控制软件被集成在同一台星载机上运行,所以从功能上看,星载机上的软件包括以下功能:①姿态控制:姿态部件 I/O 操作、定姿、姿态计算、飞行模式控制;②数据通信:星内通信、星地通信(遥测、遥控功能);③故障诊断与处理;④星载机管理:系统初始化、内存管理、时钟管理、看门狗管理、双机切换等。

系统有 4 个下位机:通信控制器、电源控制器、GPS 控制器和有效载荷控制器。各下位机上的软件主要功能是:监控分系统、采集遥测参数与星载机通信等。

根据星上软件的不同功能,可以建立小卫星软件的基本类库,见图 2。小卫星软件的基本类库

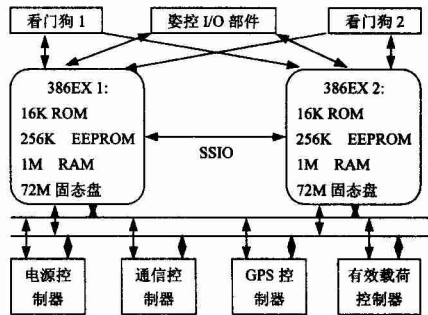


图 1 星上计算机网络系统

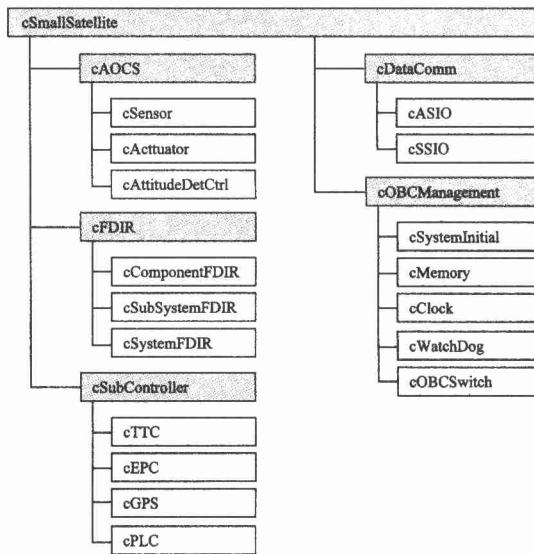


图 2 小卫星基本类库

(cSmallSatellite) 包含 5 个基本“类”：姿控类 (cAOCS)、故障诊断与处理类 (cFDIR)、数据通信类 (cDataComm)、星载机管理类 (cOB-Management) 和下位机控制软件 (cSubController)。各“类”还可以进一步细分, 具体设计见下一节。

3 基本类的设计

3.1 姿态控制类 (cAOCS)

姿态控制软件的基本功能包括各种传感器的数据采集、定姿与姿控计算、姿态控制部件操作等。姿态控制类 cAOCS 将姿态控制有关的数据和操作封装在一起, 其层次关系见图 3。

基类 cAOCS 定义了姿态控制软件公共的数据及其相关操作, 如可调的系统参数、部件的开关状态等。

传感器类 cSensor 定义各种传感器的数据采集和转换操作 GetValue(); 根据传感器与星载机接口类型的不同, 分为 8252 串口类 (c8252Serial) 8254 定时器类 (c8254Timer) 及 AD 采样类 (cADSampling)。

姿控执行部件类 (cActuator) 定义了姿控执行部件: 飞轮、磁力矩器的操作。其中包括部件的初始化、部件的控制输出 (output(ctrlVolt)) 等。

定姿和姿控计算类 (cAttitudeDetCtrl) 是姿控软件的核心, 根据卫星的飞行模式可进一步派生出 6 个“子类”: 速率阻尼模式 (cRateDampingMode), 对日捕获模式 (cSunCaptureMode), 对日定向模式 (cSunPointMode), 对地对地机动模式 (cSunToEarthMode), 对地定向模式 (cEarthPointMode), 对地对日机动模式 (cEarthToSunMode)。方法 calculate(char *inputPara, char *outputPara) 根据输入参数 (inputPara) 不同, 进行定姿和姿控计算, 并将计算结果通过 outputPara 输出。

3.2 数据通信类 (cDataComm)

星载机和下位机组成一个星上网络系统, 根据通信设备的不同, 可分为星载机与下位机之间异步串口通信 (cASIO) 和两台星载机之间的高速同步串口通信 (cSSIO)。尽管这两种数据通信方式很不相同, 但它们都涉及到对设备的初始化, 对数据帧的操作等公共部分。因此, 基类 (cDataComm) 可定义如下:

```
class cDataComm
{
    int framelength;
    char frame[framelength];
public:
    void Initial();
    void CreateFrame();
    void SendFrame(int, int);
    void RecFrame(int, int);
    void DeComposeFrame(char *);
    bool CheckFrame();
    void Communicate();
}
```

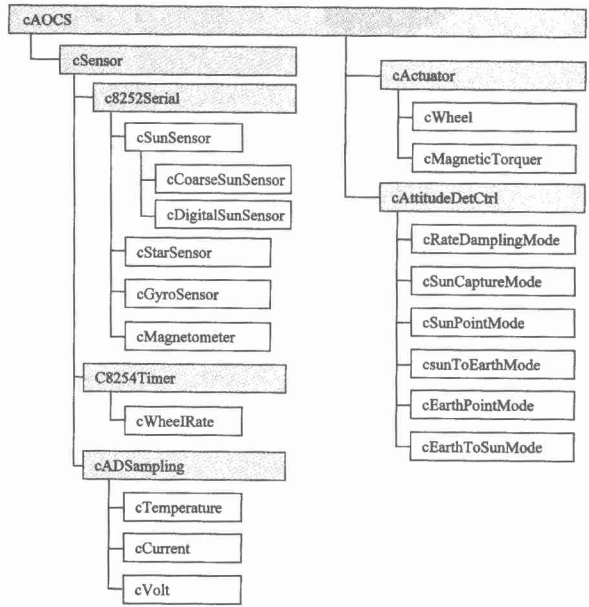


图 3 姿态控制类 cAOCS

由基类 `cDataComm` 派生出的异步串口通信类 (`cASIO`) 和同步串口通信类 (`cSSIO`) 所实现的功能不同。`cASIO` 提供星载机与下位机通信功能,星载机采集下位机的遥测参数,或转发遥控命令。当星载机与通信控制器通信时,又可进一步分为星内通信和星地通信。不论是星内通信还是星地通信,方法 `SendFrame` 和 `RecFrame` 不必改造,而在组帧 (`CreateFrame`) 时声明是星内或星地即可。星内或星地通信的不同,对帧的处理方法 (`DeComposeFrame`) 很不一样,当星内通信时,帧为通信控制器的遥测帧,当星地通信时,帧为地面站发出的遥控命令帧,此时可以采用函数重载的技术,实现对星内帧和星地帧的不同处理。

同步串口类 (`cSSIO`) 提供双机之间通信功能,帧的内容包括重要参数(如飞行模式字、星上时等)及程序段(实现双机程序对接功能)。但不论具体内容如何,其基本功能都可由基类 `cDataComm` 提供。

3.3 故障诊断与处理类 (`cFDIR`)

故障诊断与处理功能是星上软件重要的组成部分,由于空间环境恶劣,飞行器在空间可能出现各种各样的故障。试图设计一个完美的故障诊断与处理系统是不现实的,因为一个功能齐全、结构复杂的诊断系统本身就是一个故障源。因此,在故障诊断与处理的功能设计上,力求简单、有效。一般的做法是将故障分为部件级、分系统级和系统级3个级别,部件级和分系统级的故障由各分系统自行诊断与恢复,并将处理结果通知星务管理系统;而系统级故障必须由星务管理系统处理,必要时还得借助星载机硬件设备(如看门狗等)来处理。但不论是什么级别的故障,都有共同的处理过程,即参数采集、参数检测、选取故障(最高优先级故障)和故障恢复。因此,基类 `cFDIR` 可定义如下:

```
class cFDIR
{
    char *paraTable;
    char *failureRecoveryTable;
public:
    void ParaSampling();
    void ParaCheck();
    void FailureSelect();
    void FailureRecovery();
    void Process();
}
```

由基类可派生出部件级故障处理类 (`cComponentFDIR`)、分系统级故障处理类 (`cSubSystemFDIR`) 和系统级故障处理类 (`cSystemFDIR`),他们根据故障级别的不同,处理方法也有所不同。

3.4 星载机管理类 (`cOBCManagement`)

星载机管理类提供对星载机资源的有效管理,其中包括系统初始化、存储管理、时钟管理、看门狗管理和双机切换等功能。作为基类, `cOBCManagement` 只封装与星载机资源有关的参数及其操作,而派生的各个子类分别对应于不同的系统资源。

类 `cMemory` 是存储管理,在这一抽象中,它只封装存储器的大小、状态等共性信息。根据存储器的不同种类,可进一步派生出4个子类: `cPROM`, `cEEPROM`, `cRAM`, `cFlashMemory`。对于 `cPROM` 只有读操作,而其他3个子类都有读、写和刷新等操作。

类 `cClock` 是时钟管理,它提供读取星上时和设置星上时等操作;

类 `cWatchdog` 是看门狗管理。每台星载机都带有一个硬件看门狗,它被设置成一个递减的计数器。当程序正常运行时,每个执行周期的开始和结束都要喂狗(即重新设置计数器),看门狗不会出现溢出中断。但当软件或硬件出现故障,程序无法正常喂狗,看门狗会产生中断,迫使系统复位或冷启动,直到切换它机(视具体条件而定)。因此 `cWatchDog` 的定义如下:

```
class cWatchdog : oOBCManagement
{
public:
```

```

void InitialWatchdog();
void FeedWatchdog();
void IntProc();
}

```

类 cOBCSwitch 提供双机切换功能。两台星载机采用双机协同工作模式，即在进行姿态控制时，各传感器的遥测数据作为两台星载机的输入，两台星载机同时做定姿、姿控计算，并将两台星载机的计算结果进行比对，相同则输出控制；否则禁止输出，双机自检，在必要时做控制权切换（在任一时刻，只有一台星载机有控制权）。因此，类 cOBCSwitch 提供的操作包括：系统自检、结果比对、查看控制权、设置控制权等。

3.5 下位机控制器类 cSubController

运行于下位机控制器上的软件系统归属于下位机控制器类。虽然这些软件相差很大，但它们也有共性部分：比如系统的开关状态、对系统的初始化、与星载机的通信等等，可以将它们抽象到基类 cSubController 中。作为软件共用平台，它要能够适应不同有效载荷的特殊需求。从系统的基本类库可以看出，只要更换下位机控制器类的派生类：有效载荷类(cPLC)即可，软件共用平台的其它部分基本可以不改或仅做小部分修改即可。这充分体现了软件共用平台的优势。

4 TS-1 星载机软件系统的生成

基于小卫星软件共用平台，很容易通过基本类库建立具体的软件系统。下面以 TS-1 小卫星星载机软件系统为例，介绍一个软件系统是如何生成的。

在 TS-1 小卫星中，由于姿态控制软件和星务管理软件被集成在一起。因此，星载机软件系统包括姿态控制和星务管理两部分功能，即：

```

void OBCSoftSys()
{
AttitudeCtrl();
HKSoftware();
}

```

其中，姿态控制函数 AttitudeCtrl() 包括传感器参数采集、转换、定姿姿控计算及姿控部件的控制输出；星务管理函数 HKSoftware() 的主要功能是与下位机通信和对系统的故障诊断与管理。见如下程序清单。

```

void AttitudeCtrl()
{
class cCoarseSunSensor css;
class cDigitalSunSensor dss;
class cStarSensor star1, star2;
class cGyroSensor gyro1, gyro2, gyro3, gyro4;
class cMagnetometer mMeter;
class cAttitudeDetCtrl ACS;
class cWheel wheel1, wheel2, wheel3, wheel4;
class cMagneticTorquer mTorquer1, mTorquer2,
mTorquer3;
double inputPara[9], outputPara[7];

inputPara[0]=css.GetValue();
inputPara[1]=dss.GetValue();
inputPara[2]=star1.GetValue();
inputPara[3]=star2.GetValue();
inputPara[4]=gyro1.GetValue();

```

```

inputPara[5]=gyro2.GetValue();
inputPara[6]=gyro3.GetValue();
inputPara[7]=gyro4.GetValue();
inputPara[8]=mMeter.GetValue();
ACS.calculate(inputPara, outputPara);
wheel1.output(outputPara[0]);
wheel2.output(outputPara[1]);
wheel3.output(outputPara[2]);
wheel4.output(outputPara[3]);
mTorquer1.output(outputPara[4]);
mTorquer2.output(outputPara[5]);
mTorquer3.output(outputPara[6]);
}

```

```

void HKSoftware()
{
class cASIO withTTC,withEPC,withGPS,withPLC;
class cSystemFDIR sysFDIR;
}

```

```

withTTC.Communicate();
withEPC.Communicate();
withGPS.Communicate();
withPLC.Communicate();
sysFDIR.process();
}

```

不难看出,基于小卫星的基本类库,一个软件系统是很容易生成的。若每一个基本类都是经过反复验证的,则生成的软件系统的可靠性在底层得到保证。软件调试或测试人员只需把精力集中在接口和整体测试上。

对于不同的小卫星,所采用的敏感器部件或控制部件可能不同。若在基本类库中不能直接生成它们的对象,可利用一个相似的类派生具体部件的类。由派生类可直接生成这些部件的对象。当然,派生类的各项功能必须经过认真测试。每一个小卫星的设计实际是对基本类库的补充和丰富。当基本类库足够全时,小卫星的软件设计将成为一项简单快捷的工作。这也是软件共用平台的魅力所在。

5 结 论

小卫星优势之所以得以发挥,一个很重要的因素,就是充分开发和利用软件功能,使小卫星具有较强的自主飞行和自主管理功能。受到小卫星共用平台技术的启发,本文在面向对象的程序设计方法的基础上,提出了软件共用平台的技术,并以 TS-1 小卫星为应用背景,讨论了该技术的合理性和可行性。实践表明,软件共用平台技术能够高效、快捷开发小卫星的软件系统。

参 考 文 献:

- [1] A. Ginati, M. Fuchs. Earth Observation Program with Small Satellite in OHB-System [C]. Berlin: Small Satellites for Earth Observation, 2nd Int. Symp. of the Int. Academy of Astronautics (IAA) 1999,8: 12-16.
- [2] 严拱天. 萨里大学小卫星的成功经验及控制系统的一些关键技术 [C]. 北京航天工业总公司小卫星讨论组, 1996.
- [3] 何慧婉. 小卫星的发展应用前景 [C]. 重庆, 中国空间科学学会空间探测专业委员会第 9 次学术会议, 1996.
- [4] 林来兴. 当今小卫星发展水平及其关键技术 [C]. 北京航天工业总公司小卫星讨论组, 1996.
- [5] 王燕. 面向对象的理论与 C++ 实践 [M]. 北京: 清华大学出版社, 1997.

Common software platform design for small satellite

LIAO Ming-hong, WU Xiang-hu, CHENG Guang-ming

(Computer Science Department, Harbin Institute of Technology, Harbin 150001)

Abstract: For suiting common platform of small satellite, a common software platform is designed based on the object-oriented programming (OOP). Using the encapsulation, inheritance and polymorphism of OOP, a foundation class library of small satellite software is designed, the detail design of all base classes is introduced. As an example, an onboard software system is generated from the foundation class library. Experimental shown, the common software platform is an effective technique for the development of small satellite.

Key words: small satellite; software platform; object-oriented programming; housekeeping software; attitude control software