基于对象的嵌入式实时软件需求模型

徐晓东[1] 饶云华[2]

(重庆信科设计有限公司 重庆400065)[1] (华中科技大学电子科学与技术系 武汉430074)[2]

Object-Oriented Requirements Model of Embedded Real-Time Software

XU Xiao-Dong[1] RAO Yun-Hua[2]

(Chongqing Information Technology Designing CO.,LTD,Chongqing 400065)[1]

(Dept. of Electronic Sci.&Tech., Huazhong Univ. of Sci.&Tech., Wuhan 430074)[2]

Abstract The embedded real-time software requirements are analyzed, and an object-oriented software requirements model is proposed. At the same time, an example employing this requirements model is introduced in practice.

Keywords Object-oriented, Real-time system, Requirements model, Software requirements

1. 引言

软件工程就是在软件开发过程中使用一定的方法和技术来指导软件开发,尽量用较少的花费来达到所要求的目标。软件生命周期是软件工程中非常重要的一个概念,是指从分析、设计、编码到测试的整个软件开发过程。软件生命周期对于实施软件工程,开发高质量代码具有指导性的作用。

软件需求分析是软件生命周期里十分重要的一步,需求分析阶段失误常常会在后续开发阶段中被放大,所带来的损失常常是后续阶段错误所产生损失的200倍,甚至更多,所以需求分析的准确性与完备性对于整个软件的成功开发非常关键。

随着软件系统复杂性的提高和规模的增大,需求分析的作用越来越大,同时也越来越困难。如何快速地获取和准确地表达用户需求成为大型软件开发首先要面临的难题。一方面软件系统本身具有一定的复杂性;另一方面由于软件开发者对软件应用领域不了解,只能被动地接受用户提供的信息,而用户需求不全,经常变化,并且还可能对用户描述产生错误理解,因而得出不适当的需求模型;同时用户通常不知道如何按软件开发的要求来描述需求,并且用户一开始常常对自己的需求仅有一个模糊的认识,如果没有任何提示和引导,不可能一下子给出正确而且完整的需求描述。因此,如何克服上述困难,有效地取得用户需求越来越引起软件工程界的重视,并且进行了许多需求分析方法的研究[1~4]。

传统的功能需求模型存在功能的易变性、功能分解结构的随意性以及功能结构与现实问题结构不能对应等问题,而采用面向对象OO(Object-Oriented)的建模方法在解决这些问题方面较功能需求模型显示出较好的优越性。

一般认为需求分析的核心是需求获取,需求模型建立的过程就是需求获取的过程,所以对象化的需求分析的过程也就是建立基于对象的需求模型的过程。

近年来嵌入式系统已经广泛应用于各种电子设备,并且规模和复杂程度越来越大,同时嵌入式实时软件本身所具有与外部硬件相关、实时限制和代码的高效性等特点使得其开发模式与一般软件开发有所不同,所以受到越来越多的关注。

本文从嵌入式实时软件的特点出发,提出了基于对象的需求模型,详细探讨了这一模型的建立过程,并且结合一个基于pSOS+系统的机顶盒嵌入式实时软件开发介绍了这一需求建模过程。

2. 嵌入式实时系统模型及其软件设计特点

嵌入式实时软件是一个基于任务的高质量、高可靠性的实时软件系统。一个实时系统的实时性能决定于多个任务之间的相对速度和单个实时任务的绝对执行速度。和单任务系统、并行多任务系统解决的是与执行速度无关的数学领域问题不同,实时多任务系统解决的是物理领域的问题。所以,实时系统的处理模型是对客观现实的优先级的反映。

嵌入式实时系统的一般控制模型如图1所示。该系统模型由4个部分组成:被控对象、控制输出模块、控制输入模块和实时软件系统。控制输入模块主动或被动从被控对象获取信息,经处理或变换后作为输入送到实时软件进行相应处理,实时软件在规定的时间要求内将处理结果送往控制输出模块并控制被控对象的动作。

从图1可以看出实时软件与外部硬件环境关系非常密切,为了使软件的扩展性和可维护性更好,应该尽量减少对硬件的依赖。另外,嵌入式软件在CPU中是以任务的方式在运行,所以要将系统处理框图转化为多任务流程图,对处理进行任务划分。

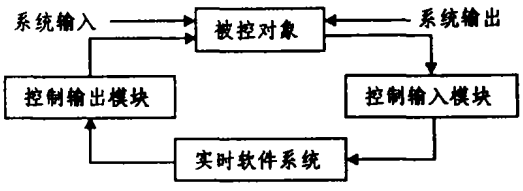

图1 嵌入式实时系统模型

任务划分存在这样一对矛盾:如果任务太多,必然增加系统任务切换的开销;如果任务太少,系统的并行程度降低,实时性就比较差,所以在任务划分时必须注意以下原则:不同的外设不同任务,因为CPU的操作快于I/O操作,如果将I/O

徐晓东 工程师,主要研究兴趣为多媒体通信。饶云华 博士研究生,主要研究兴趣为多媒体通信、计算机网络性能分析与VLSI设计等。

操作串行则会使系统执行的速度受限于I/O速度而不是CPU的速度;隔离依赖于设备的任务,也就是说非同步设备需要独立的执行时间,是ISR(Interrupt Service Routine)而不是任务,使用同步/通讯机制协调设备的ISR和相关的任务;突发事件的优先级等价于事件的时间耗尽线,不同优先级的处理具有不同任务,还需注意的是不同时间耗尽线有不同的任务,必须确认CPU的周期能够赶上所有任务的时间耗尽线,使优先级保证任务能够及时运行;将大量运算归为一个任务,通常大量运算的时间耗尽线比较长,一般使用低优先级防止它独占CPU,并且使用时间片轮询;对于周期时间,针对不同频率的处理使用不同任务,使每一个活动的任务都不依赖于其他的频率。

3. 基于对象的嵌入式实时软件需求模型

基于对象的嵌入式实时软件需求建模就是要根据实时系统的特点获取所需要的软件需求的过程。

3.1 对象的定义

对象需求模型是以OO方法,通过对象及其相互关系来表达软件需求的一种模型。这种模型能较好地克服传统功能需求模型所存在的易变性、随意性以及功能结构与现实问题结构的难对应性等问题。

我们定义基于对象的嵌入式实时软件需求模型包含对象名、操作、资源、对象关系和对象说明5个部分,如图2所示。这样整个系统的软件需求就可以由多个对象及其相互关系来表达。

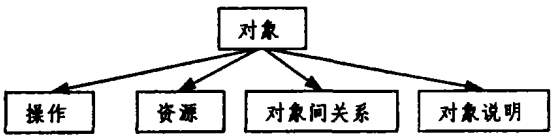

图2 面向对象的嵌入式实时软件需求模型

对象名是对象的名称,是对象的唯一标识,不同的对象具有不同的对象名;

操作是指对象相关的所有任务,可以改变对象的状态;

资源是指对象进行操作的条件,主要是与实时调度有关的时间片切换、中断和任务优先级等资源描述;

对象间关系描述本对象与其它对象的联系。虽然对对象的定义都是相对独立的,但对象间实际存在着一定联系,这样有可能导致对象需求描述的不一致性。通过对象间关系可以很方便地对需求进行一致性检查。

对象说明主要定义了与对象操作有关的数据结构。

3.2 对象的划分

嵌入式实时软件需求对象的划分要根据系统的实际情况来进行。为了使软件具有良好的扩展性、高效性,并满足系统要求,在划分对象的时候必须注意以下几点原则:

1)对整个系统进行分析时,由于嵌入式系统平台采用事件驱动的方式进行任务调度,从系统的角度去看,任务是参与竞争系统资源的最小单元,因此相应的对象划分也以任务为最小单元来进行;

2)在对象划分过程中,尽量减小对硬件和操作系统的依赖程度,采用硬件抽象层可以增加软件的硬件无关性;操作系统抽象层用于实现操作系统层的可移植性,以支持多种RTOS(Real-Time Operating System);

3)在对象划分过程中,尽量保证对象之间的相互独立,以避免对象之间出现重叠和不一致性,使对象之间的关系清晰明确,可以减轻需求模型检测的工作量,并使后期设计和实现更加容易和方便;

4)当对象较多时,可以合并或删除一些对软件系统并不十分重要的对象,减少对象数量,减轻需求定义的工作量。

根据上述规则并考虑实际系统的特点,在对硬件和操作系统进行抽象的基础上,得到了具有开放性和可扩展性的机顶盒应用层软件系统需求模型,其用户接口(GUI)、EPG、资源管理、业务调度、业务解释和呈现等主要对象划分如图3所示[5,6]。

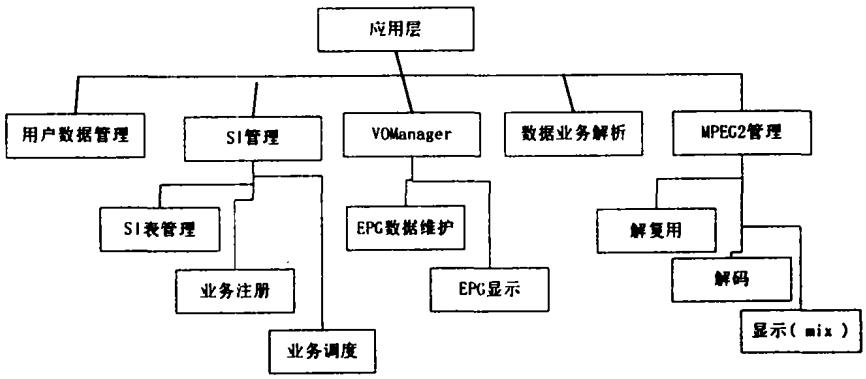

图3 机顶盒实时软件对象划分实例

3.3 对象的描述

对象的描述是在对象划分的基础上对每个对象进行详细定义的过程。

对象的操作主要是指与对象功能相关的任务等,这里要优先考虑任务之间的耦合关系;另外还必须分析实时系统的时间耗尽线,要着重注意以下几个方面的问题:中断和上下文切换的表述;多任务和多进程所产生的并发活动;任务间的通信和同步;数据和通信的突发性;时间约束的表述;异步事件和任务的处理。

对象的资源主要就是与调度控制有关信号量、消息和事件以及任务优先级等。

对象说明主要是对象操作有关的数据结构的定义。

对象间关系的描述对于需求模型的一致性检测非常重要,主要有如下关系:

1)connect_to(对象名,操作名;对象名,操作名;数据名)

2)connect_from(对象名,操作名;对象名,操作名;数据名)

3)part_of(对象名;对象名)

4)part_from(对象名;对象名)

5)overlap_with(对象名,资源名;对象名,资源名)

其中对象名和操作名指对象的操作。Connect_to 表示第一个参数所表示的对象操作输出数据到第二个参数所表示的对象操作,而 Connect_from 则正好与之相反;part_of 表示第一个参数所表示的对象是从第二个参数对象继承而来,part_from 则与之相反,这个关系主要是说明对象之间的继承关系;overlap_with 表示第一个参数的资源与第二个参数的资源有重叠。

例如采用了面向对象需求分析方法的机顶盒软件设计中两个最重要的 SI 管理对象和用户输出管理对象(VOManager)的定义如下。

SI 管理对象主要是从 DVB-SI(Digital Video Broadcast-Service Information)数据缓冲中获取各种 SI 表,建立起整个导航数据结构;发送 EPG(Electronic Program Guide)数据到 EPG 数据缓冲;接收用户的消息,对用户的请求进行处理。(包括菜单浏览、节目预定等)等。

```
SI_Manager{
Operation: SVCManager; //控制从 MPEG 流中提取 PSI 表.
           DMXSection_CopyTask; //接收私有分段回调函数的信
                                  号,执行 copy 任务.
           SKManager; //接收所维护的 socket 上收发的消息.
           …
Resource: priority_of_ SVCManager = 180; // SVCManager 的优
                                          先级为180.
          priority_of_ DMXSection_CopyTask = 190; // SVCM-
                                          anager的优先级为190.
          …
Relation: connect_to(SI_Manager,SI_reponse;VOManager,Send_
          Init_SIMgr;SI)
          …
Exponents: PAT; //程序关联表.
           CAT; //条件访问表.
           PMT;//程序映射表.
           TSDT;//为支持多种基于 MPEG-2的系统.
           EIT;//网络信息表--当前.
           SDT;//服务描述表--当前.
           EIT;//事件信息表-当前:现在/下一个.
           TDT;//时间描述表.
           …
           }
```

用户输出管理对象(VOManager)接收并解释用户输入数据,对用户授权信息进行认证,对用户的个性化数据进行管理、配置等。

```
UI_Manager{
   Operation: VOManager;// 主消息循环任务
              VOMSendEvent;// 处理各种用户事件
              Send_Init_SIMgr;// 向 SI 管理器发送启动命令,获
                                取 SI 信息
              …
   Resource: priority_of_ VOManager = 170;
             SmartCard 事件;
             …
   Relation: connect_from(SI_Manager,SI_reponse;VOManager,
             Send_Init_SIMgr;SI)
             …
   Exponents: OSD_MENU ;//菜单链
              OSD_GENOBJECT;// 所有 GUI 对象的抽象
              …
}
```

3.4 一致性检测

一致性检测是检查、处理或解决对象内部和对象间的不一致性的过程,下面我们主要讨论对象间的不一致性处理过程。

对象间的一致性检测主要有以下几个规则:

R1:connect_to(X,A;Y,B;d)→connect_from(Y,B;X,A;d)

R2:part_of(X;Y)→part_from(Y;X)

R3:overlap_with(X,A;Y,B)→overlap_with(Y,B;X,A)

R1规则表示对象 X 中的关系 connect_to(X,A;Y,B;d)必然与对象 Y 中有一条关系 connect_from(Y,B;X,A;d)相对应,反之,对象 Y 中的关系 connect_from(Y,B;X,A;d)必与对象 X 中有一条关系 connect_to(X,A;Y,B;d)对应。R2和R3的规则和 R1类似,这里就不再说明。

例如在 SI 管理器对象中有一条 connect_to(SI_Manager,SI_reponse;VOManager,Send_Init_SIMgr;SI),那么在 UI 管理器对象中就应该相应的关系与之对应 connect_from(SI_Manager, SI_reponse; VOManager, Send_Init_SIMgr;SI),否则这两个对象间就存在不一致性问题。不一致性检测的步骤如下:

1. 根据规则检查对象间的关系,如果不存在不一致性,则检测通过,否则转到下一步;

2. 检查具有不一致性的对象,若对应的对象不存在,则重新进行对象划分并进行需求描述;

3. 检查操作等部分的不一致性,若存在不一致性,则重新考虑对象间的任务划分,解决不一致性问题;

4. 重新整理两个对象间的关系,使其符合一致性规则,然后转到1。

结论 面向对象的方法由于其本身固有的优点在软件开发的很多领域都得到了应用,本文介绍了一种基于对象的嵌入式实时软件需求分析模型,该模型能够提高系统的重用性、可维护性、可靠性和可扩展性。并且我们将这一方法应用于综合业务机顶盒的开发,取得了良好的效果。

一个好的开发模型需要经过大量的应用实践和深入的分析研究才能不断地成熟,所以,下一步还需要继续对该方法进行理论和实践两方面的补充和完善,使该模型能够更加适合嵌入式实时软件的需求分析。

参 考 文 献

1 Nuseibeh B, Krammer J, Finkelstein A. A Framework for expressing the relationships between multiple views in requirements specification. [J] IEEE Trans. On Software Engineering,1994,20(10):760～773

2 Blyth A J C, Chudge J, Dobson J E, et al. A Framework for Modelling Evolving Requirements. [C] IEEE Computer Software and Applications Conference, 1993. 83～89

3 Thompson J M, Heimdahl M P E, et al. Specification-Based Prototyping for Embedded Systems. [J] Software Engineering Notes,1999,19(5):163～179

4 Jaffe M S, Leveson N G, Heimdahl M P E, et al. Software requirements analysis for real-time process-control systems. IEEE Transactions on Software Engineering, 1991,17(3):241～258

5 饶云华,徐重阳,刘政林. 基于 DOCSIS 的视频点播. 电视技术,2001(9):48～50

6 刘小卫. 嵌入式软件在综合业务机顶盒上的应用研究:[华中科技大学硕士论文]. 2001