

基于 UML 的嵌入式实时控制系统的建模与实现

王小平¹, 宣乐飞^{1,2}, 张 蔚¹

(1. 杭州电子科技大学 计算学院, 浙江 杭州 310018;

2. 杭州职业技术学院 信息电子系, 浙江 杭州 310018)

摘 要:随着嵌入式系统在各个领域的广泛应用, 嵌入式系统变得越来越复杂, 研究一种支持嵌入式系统从分析、设计、验证到编码这一整个过程的模型系统及建模方法变得越来越重要。用 UML 对嵌入式系统进行面向对象分析与设计, 并以电梯实时控制系统为例, 建立系统的静态模型和动态模型。在 IAR visualSTATE 环境下, 分析系统状态, 实现了从状态图到 C 代码的自动生成, 这提高了软件设计和实现的效率、质量、维护性和扩展性。

关键词:标准建模语言; 嵌入式系统; 实时控制; IAR visualSTATE

中图分类号: TP302.1

文献标识码: A

文章编号: 1673-629X(2006)07-0239-03

UML - Based Model Design and Realization in Embedded Real - Time Control System

WANG Xiao-ping¹, XUAN Le-fei^{1,2}, ZHANG Wei¹

(1. Sch. of Computer, Hangzhou Dianzi Univ., Hangzhou 310018, China;

2. Dept. of Info. and Electronics, Hangzhou Vocational Technology College, Hangzhou 310018, China)

Abstract: Embedded system becomes more and more complicated with its widespread use in a variety of domains. Methods of analyzing, developing, modeling and verifying embedded system are increasingly important. This paper introduces the process of object-oriented analyzing and designing an embedded system with UML, establishes its static model and dynamic model by taking the elevator real-time control system as an example. It analyzes system states and transforms state diagrams into C code automatically in IAR visualSTATE. The efficiency, quality, maintenance and expansibility of software designing would be improved based on this method.

Key words: unified modeling language; embedded system; real-time control; IAR visualSTATE

0 引言

结构化设计方法是目前使用广泛的嵌入式软件设计方法, 但设计出来的软件在复用性、扩展性、维护性等方面不尽人意, 主要原因在于结构化方法将数据和在数据上的操作分离开来, 对象封装性和信息隐藏性不足。同时, 由于在开发过程中, 高层次的系统分析抽象阶段和目标环境的开发编程阶段几乎没有中间步骤, 传统的嵌入式软件开发方式已不再满足当前的发展趋势^[1,2]。

面向对象方法基于信息隐藏、类、继承的概念。信息隐藏可以带来更好的自包含, 继承提供了一种系统化改编类的途径。引入面向对象分析与设计的方法能有效解决传统软件过程方法出现的问题^[3]。UML (Unified Modeling Language) 是面向对象技术领域内占主导地位的标准建模语言, 主要作用是帮助用户对软件系统进行面向对象的描述和建模, 描述了从需求分析到实现和测试的软件开发全过程。

1 基于 UML 的嵌入式实时软件建模

1.1 UML 简介

UML 已经成为一个得到包括 Compaq, IBM, Intel, Microsoft, Sun 等众多知名大公司在内的广泛接受的标准, 被 OMG 组织强烈推荐为面向对象的产品研发过程中标准的建模语言。UML 是一种通用可视化建模语言, 包括 UML 语义和 UML 表示法两部分, 它结合 Booth, OMT, Jacobson 方法的优点, 统一符号体系, 并从其它方法和工程实践吸收了许多经过实际检验的概念和技术^[4]。

1.2 用 UML 进行软件建模的步骤

软件过程可分解为若干活动, 活动与活动之间存在复杂的时序关系, 如同步、异步、反馈、串行、并行等。建模是通过将用户需求映射为代码, 保证代码满足这些需求, 并能方便地回溯需求。当采用 UML 建模时, 第一步是描述过程需求, 即进行过程分析。主要是识别角色, 识别用例, 建立角色与用例之间的关系, 得到用例图。第二步, 用类图描述出过程模型中的重要元素 (如产品、活动等) 之间的静态关系。第三步, 描述活动、产品等元素在执行时的时序关系或交互关系, 动态行为用顺序图和状态图描述。已

收稿日期: 2005-09-24

作者简介: 王小平 (1979-), 男, 福建龙海人, 硕士研究生, 研究方向为智能嵌入式系统。

经建立的过程模型在过程模型环境的支持下是可执行的, UML 描述被转化为代码。可执行的过程是过程模型的实例。通过对过程执行的跟踪和监控, 收集与过程有关的量度, 对过程进行评价, 以便改进过程模型^[5]。

2 电梯实时控制软件的静态建模

2.1 系统描述

选用电梯系统作为论述 UML 建模的一个实例, 因为它作为一种典型的嵌入式实时系统更容易被人们理解和接受。与现实的电梯系统相比, 笔者忽略一些无关的细节, 只是抽象了一般的电梯系统都有的基本功能和操作。

假设电梯在一幢大楼的第一层到第 MAX_FLOOR 层之间移动, 当电梯停在某一层, 电梯当前运行方向的标识灯点亮, 乘客获得当前电梯运行的方向。电梯在两个楼层之间快速移动, 但它应该能提前减速停在目的层。乘客通过观察电梯移动方向和电梯位置标识器决定是否进/出电梯。

2.2 用例图

用户都是为了某种目的与系统的自动化角色进行交互, 人和角色都希望系统的行为是可预知的。在 UML 中, 用例图描述一组用例、角色和它们的关系, 是一个系统行为、一个子系统或一个类建模的中心。基于电梯控制系统的需求文档, 共有 1 个角色——乘客 (Passenger) 和以下 3 个用例:

- 标识运行方向 (Indicate Moving Direction): 电梯应该有这种机制, 即让乘客知道电梯目前的运动方向, 决定是否进电梯。

- 电梯的动/停 (Move/Stop the Drive): 这是一台电梯的主要功能, 详细的动作包括驱动速度的改变、停止的判定、电梯的运动方向驱动。

- 标识电梯位置 (Indicate Drive Position): 类似地, 电梯应该让乘客知道他/她的目的层是否到达, 决定是否离开电梯。

2.3 类图

类图是面向对象系统中应用最广的静态图, 用来描述一组类、接口和协作, 及它们的关系。通常用类图来描述整个系统 (比如类的属性和操作等) 和建立静态结构 (比如系统的结构和细节等)。在对电梯系统用例的描述的基础上, 从系统对象组成的角度建立以下 4 个类:

- 电梯控制器 (Elevator Control): 电梯系统的核心控制对象。

- 电梯 (Drive): 电梯在电梯控制器的控制下以不同的速度上升和下降, 需要时可以停下。

- 按钮 (Button_Count): 是电梯控制器类 (Elevator Control), 也是控制按钮类。它控制对象与按钮对象的通信, 得到按钮是否被按下, 并比较当前楼梯的位置和所要

到达的楼梯的位置。

- 标识器 (Indicator): 提供电梯的当前位置和移动方向, 系统有两类标识器: 电梯位置标识器 (Elevator Position Indicator) 和电梯方向标识器 (Elevator Direction Indicator)。

软件的架构视图可以解决对象构造视图中出现的一些问题, 如: 当超过一个对象想同时得到中央控制对象的控制时, 这些对象竞争控制器有限的计算资源是不可避免的, 一些对象不能及时得到维持正常运行的控制消息。

电梯系统的软件架构如图 1 所示: 控制任务被分配到几个控制对象中, 每个控制一个或两个环境对象 (被控制对象所控制的对象被定义为环境对象, 尽管它们存在于电梯系统, 但不属于软件控制系统), 都没有负担过重或空闲。除了 Dispatcher, 所有其它的控制对象都是从超类——电梯控制器继承而来。这些控制对象共享电梯控制器的一些属性, 而且有用其控制对象的自己属性和方法。

以下是图 1 的部分说明:

- 驱动控制器 (Drive Control): 控制电梯驱动, 它使电梯上下移动, 在需要时停下。

- 电梯按钮控制器 (Button Control): 电梯按钮控制器接受电梯呼叫按键的呼叫, 并控制相应的电梯信号灯的开关。

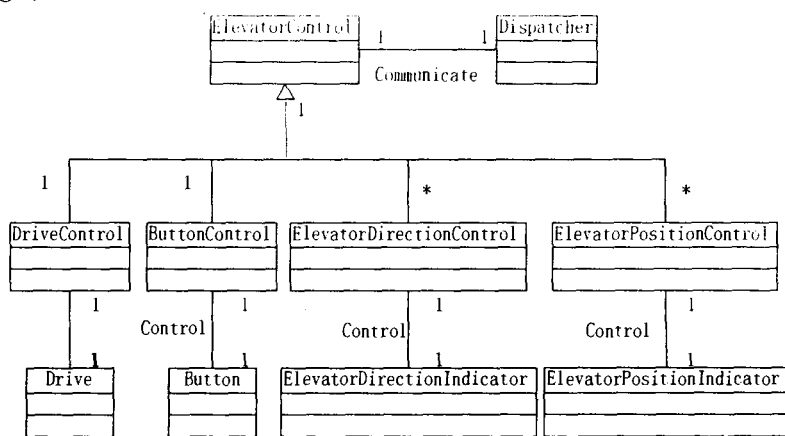


图 1 类图——软件架构图

3 电梯控制系统的动态建模

UML 建模是一个可裁剪的过程, 使用者不必面面俱到地画出各种图。对于每一幅图, 只有在必要时 (比如能帮助分析、设计、指导编码、加深理解、促进交流等) 才需要画出, 这样的图对建模才有意义。UML 提供顺序图和协作图对系统进行动态建模, 由顺序图能较容易得到协作图。限于篇幅, 文中只给出电梯系统软件的部分顺序图。

3.1 处理电梯呼叫的动态模型

“处理电梯呼叫”这个用例有 2 个场景。场景 1: 电梯呼叫服务——电梯向乘客的目的地的方向移动; 场景 2: 电梯呼叫服务——电梯向乘客的目的地的反方向移动。这 2 个场景可以共享同一个用例。这两个场景的顺序图

如图 2 所示。

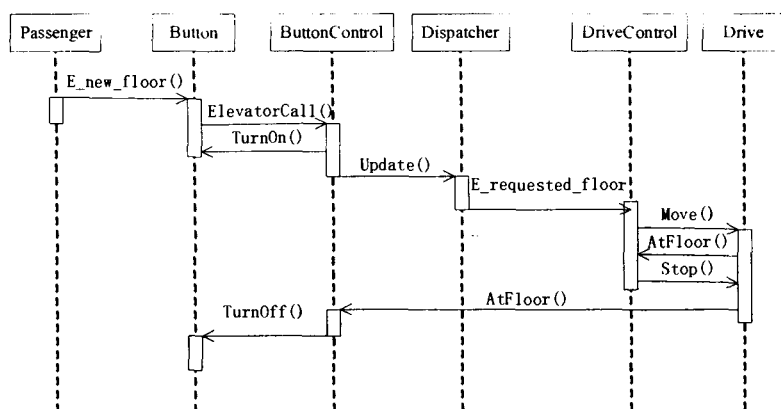


图 2 电梯呼叫服务的顺序图

3.2 电梯系统的状态图

状态图显示了通过正向工程和逆向工程构造的可运行的系统,其中的状态机实现对象的行为建模。对象的行为可以响应上下文之外分发来的事件。对象有一个清楚的生命期,它的过去影响了当前的行为。

文中举例的电梯系统是一个 4 层电梯,用 4 个 LED 灯来显示电梯的位置,用 3 个 LED 灯来显示电梯的运行方向,具体状态图的建立和转化将在下文给出。

4 基于状态图模型的代码生成

4.1 IAR visualSTATE 简介

IAR visualSTATE^[6]是 IAR 公司新近推出的一套基于状态机的与 UML 兼容的图形化软件设计环境。它支持层次和并发状态机,主要有以下几个特点:①基于状态机的 UML 兼容的图形化工具;②先进的确认和验证技术;③通过 RealLink 实现在目标上调试;④与 Altia FACE-PLATE 一起使用可以进行原型法设计;⑤自动生成独立于处理器、实时操作系统和编译器的 C 代码;⑥生成的代码非常紧凑而有效,能够适合 8 位和 16 位目标;⑦能自动生成完整的文档。

visualSTATE 为开发者提供了部分头文件和源文件,还有 API 函数。API 函数可以当作 C 源代码在编译时使用,用来初始化 IAR visualSTATE 环境中的变量、传递事件以待系统处理、调用行为函数以及改变系统的状态等。

4.2 Elevator 状态图

Elevator 状态机视图描述当前电梯楼层的状态转移,当要到达新的楼层的事件发生时,电梯向新的楼层运行,状态机也转移到一个新的楼层,并用相应的灯信号更新楼层显示。在 IAR visualSTATE 下建立的 Elevator 状态机视图如图 3 所示(由于篇幅的关系,Button 状态机视图和 Drive 状态机视图没有在文中给出)。

4.3 从状态机模型生成 C 代码

首先,在 visualSTATE 平台下建立系统的状态机视图,用“validator”从以下 3 个方面验证:①检查状态机的运

行、事件的组合是否覆盖所有状态;②检查是否有死状态,

按照行为逐项运行状态机,确定所有需求条件被覆盖;③状态机改变状态,采取行动,正确地修改变量,确定没有遗漏和冗余。接着,选择菜单 Project>Options>Code generation,配置电梯实时控制系统的代码生成选项对话框。最后,用 visualSTATE 中 Navigator 的 Code generation 来生成相关代码,其结果如图 4 所示。

由于 IAR visualSTATE 提供的代码是独立于处理器、实时操作系统和编译器的 C 代码,所以在应用过程中要选择编译器,另外还必须编写一部分和处理器相关的代码以及调用状态机代码的函数等程序。在添加必要的分代码后,选择用 IAR Embedded Workbench 作为编译器,把相关的文件放在一起编译,可以在基于 ARM 的硬件环境中模拟电梯控制系统的运行状态。

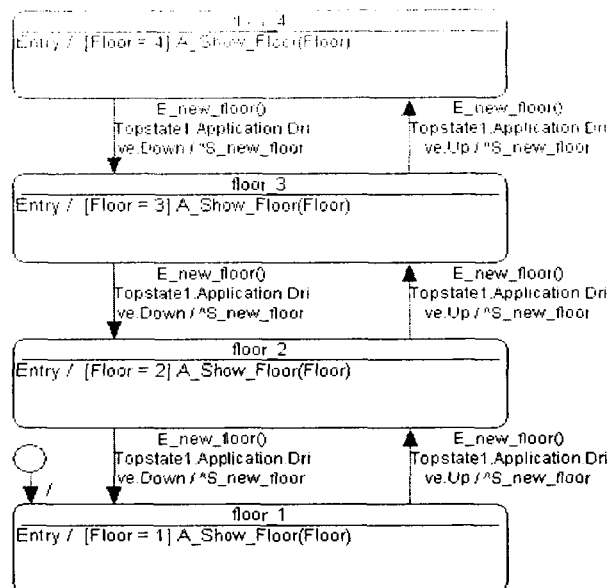


图 3 Elevator 状态机视图

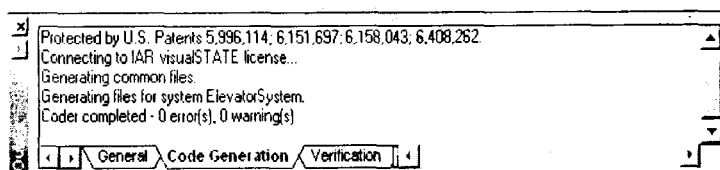


图 4 电梯系统的代码生成的输出对话框图

5 小结

介绍了用 UML 对嵌入式系统软件的建模过程,结合基于状态机的设计工具 IAR visualSTATE 及其它所提供的 API 函数,可以自动生成独立于处理器的 C 代码。实践表明:生成的 C 代码与设计绝对一致,执行是完全确定的,其代码比手工编写的更紧凑。因此,借助高层开发环

(下转第 244 页)

DOCUMENT 以分析 XML 文档并返回准备使用的已分析文档的句柄。已分析文档是 XML 档中各种节点(元素、特性、文本、注释等)的树型表示法。文档句柄传递到 OPENXML 后,OPENXML 根据传递给它的参数提供文档的行集视图。使用完毕后必须调用 SP_XML_REMOVEDOCUMENT 系统存储过程释放内存,以将 XML 文档的内部表示法从内存中删除^[4]。

总结来讲,OPENXML 的使用步骤为:①调用系统存储过程 SP_XML_PREPAREDOCUMENT 创建 XML 文档的内部表示法,并句柄传递给 OPENXML。②OPENXML 基于 XPath 提供此文档的行集视图。③调用系统存储过程 SP_XML_REMOVEDOCUMENT 将 XML 文档的内部表示法从内存中删除。

4 将 XML 文档设置为命令文本并返回结果集

SQL OLE DB 已为 XML 和 XPath 做了扩展。添加了一个 ICommandStream 接口以用来向 OLE DB 传递需要处理的模板。另外,OLE DB 提供程序的属性集也得到扩展,允许支持 IStream 的对象规范返回 XML 结果以及支持使用通过映射架构定义的 XML 视图进行 XPath 查询。

流对象是在 ADO 2.5 中引入的,现在可以与新引入的语言分支一起来向数据库发送 XPath 查询和从数据库接收 XML 结果。可以通过将 ADODB.COMMAND 的属性“Output Stream”设置为 Response 和“xml root”根节点元素,并且设置 Response.ContentType 为“text/xml”,来规定执行结果以流的形式返回。

5 在 SQL Server 2000 中使用 ADO.NET

笔者在实际的项目实践中经常遇到要处理不同的 ASP.NET 工程的情况,它们一般都是用 .NET 提供的 ADO.NET 来提供访问。前面介绍的 HTTP 查询 SQL Server 好像就显得无用武之地了。在这种情况下,其实仍然可以通过在查询中使用 FOR XML 关键字直接从数据库中检索 XML^[5]。

值得注意的是,从上述类型的查询中返回的 XML 数据格式并不是正确的。ADO.NET 的 SqlCommand 类提供了解决方法 ExecuteXmlReader()。该方法返回了一个可

以解析的 XML 的 XmlReader 对象。使用该方法可以得到格式正确的 XML,后者被简单地输入到浏览器中。通过将从 SQL Server 2000 中返回的 XML 片断动态地装载到 XpathDocument 类^[6],并使用 XSLT 对其进行了转换,相同类型的处理会变得更加有用。由于 SQL 查询可以返回多个客户记录,可以想象在将此 XML 传递给 XpathDocument 的构造函数之前,这些记录需要将被封装在一个根节点内部。幸运的是,XpathDocument 类负责自动添加一个根节点,这使得通过 XSLT 转换 XML 变得非常容易。其他应用如当从一个 FOR XML 查询中返回的数据依赖一个 DTD 的一部分时,可使用 XmlParserContext 类^[6]。

6 结 论

SQL Server 2000 提供了许多处理 XML 数据的有用特性。通过分析利用这些特性,可以很容易地从数据库中提取 XML 文档,这对基于 XML 的 Web 系统开发是非常有益的。XML 数据可以设定其样式以便在浏览器中显示,也可以交给 VBScript 或 JavaScript 进行后台处理后再显示。在该机制中,XML 文档是(分布式)数据交换的有效途径。网络中的服务器、客户机和应用程序所进行的处理均可使用这种机制交换数据。目前,基于 SQL Server 2000 XML 的 Web 开发已经迈向真正的多层模式^[7]。

参考文献:

- [1] Microsoft Corporation. Microsoft SQL SERVER 2000 XML 和 Internet 支持[M]. 北京:科学出版社,2000.
- [2] 沈兆祥,李 劲. SQL Server 2000 与 XML 整合应用[M]. 北京:清华大学出版社,2001.
- [3] 申朝阳,宋颜浩. ASP.NET 与相关数据库技术[M]. 北京:中国水利出版社,2005.
- [4] 飞思科技产品研发中心. SQL Server 2000 XML 应用开发[M]. 北京:电子工业出版社,2001.
- [5] WAHLIN D. 基于 XML 的 ASP.NET 开发[M]. 王宝良译. 北京:清华大学出版社,2002.
- [6] Otey M, Otey D. ADO.NET 技术参考大全[M]. 史创明,崔金铃译. 北京:清华大学出版社,2003.
- [7] 陶琴慧. SQL Server 2000 XML 数据交换技术在 WEB 系统中的应用[J]. 微型机与应用,2003(8):35-37.

(上接第 241 页)

境可以实现基于模型的调试和基于源程序的调试同步,这大大简化了开发难度和提高了工程效率。

参考文献:

- [1] 陈晓锋,戴国骏,洪小富. 基于 UML 的嵌入式指纹识别系统软件建模[J]. 计算机工程,2004,30(14):130-132.
- [2] 陈思功,秦 晓,章恒羽中. 基于 UML 的软硬件协同设计的模型分析方法[J]. 软件学报,2003,14(1):103-109.
- [3] 李思广,林子禹,胡 峰,等. 基于 UML 的软件建模方法

研究[J]. 计算机工程与应用,2003,39(6):76-78.

- [4] Object Management Group, Inc. Introduction to OMG's Unified Modeling Language (UML) [EB/OL]. http://www.omg.org/gettingstarted/what_is_uml.htm, 2005-05-26.
- [5] Gomaa H. 用 UML 设计并发、分布式、实时应用[M]. 吕庆中,李 焯,罗方斌译. 北京:北京航空航天大学出版社,2004.
- [6] visualSTATE User Guide Version 5[Z]. USA: IAR System Software Inc, 2002.