

基于组件技术的嵌入式电力监控系统的设计与实现

方海 吴健 张蕴

(西北工业大学计算机学院,西安 710072)

摘要 分析了嵌入式组件的特点以及当前几种著名的嵌入式组件模型的优缺点,提出了一种基于组件技术的嵌入式电力监控系统的设计方法。组件技术的应用提高了嵌入式软件开发的效率和质量,增加了系统的可扩展性和复用性。

关键词 组件 电力监控系统 组态 嵌入式软件

中图法分类号 TP311.5 **文献标志码:** A

嵌入式系统体积小,功耗低,成本低以及适应恶劣环境的特点决定了嵌入式系统在工业控制中的应用越来越广。嵌入式系统与组态软件结合起来以快速、高效、方便地开发出适应各种场合的监控程序,已成为工业控制领域发展的必然趋势。

随着嵌入式软件的规模和复杂性不断增大,对软件的可重用性、可配置性和扩展性要求越来越严格,复用是解决这些问题的有效方法。组件或构件技术是软件复用的主流方向,嵌入式系统软件开发需要基于组件的软件开发技术。

1 嵌入式组件

1.1 嵌入式组件的特点

嵌入式系统的特征与需求,决定了嵌入式组件与通用组件在结构、接口、功能/性能等方面有很大差异。嵌入式组件除了符合通用软件组件封装性、可重用性要求外,还必须能够满足嵌入式系统的固有特性,因此还具有以下特点^[1]:时间约束,它指嵌入式组件运行时对时间的限制,如开始执行时间、结束时间、执行间隔时间、执行优先级等;组件体积限制,由于嵌

入式硬件资源有限,单个嵌入式组件不能包含过多的功能,组件的体积要限制在一定范围内;和硬件设备紧密相关,和通用组件相比,在嵌入式组件开发过程中要考虑到硬件平台的差别;能运行在目标板的操作系统上(如 WINCE, Linux, VxWorks 等),且对底层硬件外围设备开销不能太大;组件生命期长,开发完毕、经过测试的组件一般会有较长的生命期;可靠性和安全性要求高,嵌入式系统对可靠性和安全性均有很高的要求,一旦代码使用不当,将有可能导致整个系统的崩溃,嵌入式组件是嵌入式系统的基本组成单位,这就要求嵌入式组件有高可靠性,为整个系统安全运行提供保证。

1.2 嵌入式组件模型

组件模型是进行组件式软件开发和组件复用的基础。当前流行的通用组件模型包括微软公司的 COM、SUN 的 EJB 和对对象管理组织的 CORBA 组件模型 CCM。嵌入式组件的特点决定了通用组件模型不能适用于嵌入式组件开发。嵌入式组件模型必须能够充分刻画嵌入式系统的固有特征。

当前主要有如下几种嵌入式组件模型:

比利时 IWT 协会赞助的 SEESCOA 项目的 CCOM 模型^[2]:组件模型中的元素包括接口(Port),接口连接器(connector)。接口有功能(role)和数量(multiplicity)两种属性:功能(role)表示接口在组件与组件之间通信时的作用;数量(multiplicity)表示

2010年11月8日收到

第一作者简介:方海(1983—),陕西人,硕士研究生,研究方向:软件工程。

组件有多少此种类型的接口。在 CCOM 模型中接口具有四层意义:①语法级(Syntactic level):描述接口中消息的名称和参数形式。②语义级(Semantic level):描述接口的前置和后置断言。③同步级(synchronization level):描述接口的消息序列、循环和替换路径等。④质量级(QoS level):描述组件的非功能性约束需求。CCOM 的组件建模中采用契约(Contract)来描述非功能型约束,且较多地使用了实时建模的方法描述时间约束,但是当前还没有引入其它非功能性约束的方法。

飞利浦公司用于消费电子的 Koala 组件模型^[3]:在 Koala 模型中,接口是最底层的设计和重用单元,从语法的层次来看,接口是相关函数的小型集合。Koala 模型中的接口包括入和出两种;组件之间通过接口连接,接口在设计时是静态连接的。由于 Koala 模型中没有考虑非功能性约束,这对于嵌入式系统应用来讲具有很大的局限性。

ABB 等公司用于现场总线技术的 Pecos 组件模型^[4]:Pecos模型中的元素包括接口(ports)、组件;接口有入(in)、出(out)、入出(in&out)三种类型;组件也有三种类型:活动组件(active component)、被动组件(passive component)和事件组件(event component)。在此模型中一个接口被映射为一个共享的变量。由于现场总线设备资源非常受限,典型配置为 16 位微处理器、256KB ROM、40KB RAM,所以此种模型在嵌入式系统中不具有通用性。

SaveCCM 嵌入式组件模型^[5]:应用于汽车电子领域,着重于实时与依赖等非功能属性。SaveCCM 主要由下述四个部分构成:①组件:封装了行为的基本单元,根据执行模型来执行;②转换器:提供动态改变组件的连接结构(可在配置时或运行时改变);③组装机:提供由组件和转换器聚合成新组件的方式;④运行框架:提供如组件间通讯、组件执行和组件控制的服务集合。与其它嵌入式组件模型相比,SaveCCM 模型更侧重于可预测性和可分析性,对模型的可配置性(灵活性)相对比较弱化。因此,该模型在一些应用方面受到限制。

2 系统功能描述

电力监控系统的设计目标是为电力部门提供便利、快捷、安全的从开发到现场应用的电力设备监控系统。系统提供可定制用户界面、采集并显示现场数据、控制现场设备、现场设备状态的显示、现场设备报警提示、设备数据的转存、电网质量分析、系统运行日志记录、设备特性曲线的动态显示、负荷分析、设备故障诊断和分析等功能。

图 1 描述了嵌入式电力监控系统的主要功能。

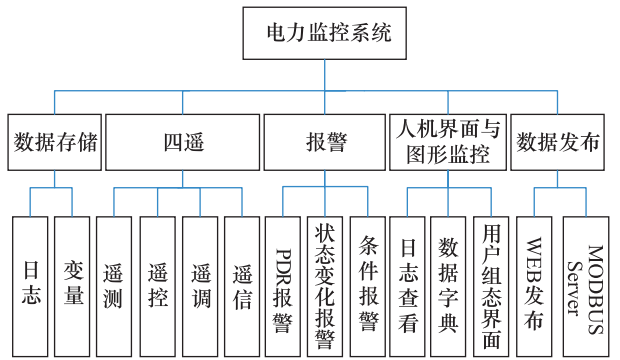


图 1 系统的功能划分

3 系统的设计和实现

3.1 系统结构及组件的划分

嵌入式电力监控系统是对已有通用电力监控组态系统(YSS2000)的一个扩充,通用系统由设备开发子系统、工程开发子系统和实时监控子系统三部分组成。设备开发子系统和工程开发子系统组成监控系统的组态开发环境;而实时监控子系统则负责对现场设备的监测和控制。设备开发子系统用于开发设备库,设备库以抽象设备为基本单位存储不同型号的设备信息,设备信息包括设备的基本信息、窗口信息和变量信息等;工程开发子系统用于开发具体的工程实例,工程实例是对监控现场一种软件模拟,工程中将设备实例化为子站,作为工程组成的基本单位。工程实例可以直接投运于实时监控子系统,对现场设备进行实时监控。工程实

例信息以 XML 形式存储在硬盘上。

嵌入式电力监控系统是实时监控子系统在嵌入式系统上的实现。嵌入式电力监控系统读入工程实例对现场中的设备进行数据监控和报警。嵌入式监控系统分为初始化和运行两个阶段,初始化阶段中系统从文件和数据库中载入用户在工程开发系统中的建立的组态数据,并构建实时数据库结构;而在运行阶段,通过启动定时器和通信线程系统完成显示的更新和数据的实时更新。

系统和现场设备之间的数据交换是双向进行的。一方面系统需要及时地读取现场设备的数据,另一方面系统需要向现场设备发送数据进行功能控制和参数调整。图 2 中的箭头代表数据的流向。系统可根据数据流划分为表示层、业务逻辑层和通信层。

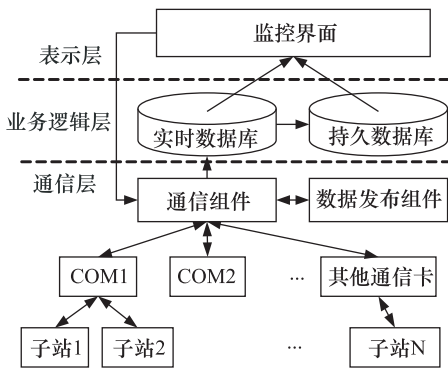


图 2 系统数据流示意图

本系统基于软件组件技术,以功能的不同划分为不同的组件,组件的功能大小决定了组件的粒度。从而达到组件高内聚低耦合的设计要求。组件化的系统结构的设计如图 3 所示。



图 3 功能体系结构

3.2 组件模型

参考 COM 模型,也使用 GUID 来标识接口和组件。每一个组件提供服务的接口 IProvided 都要继承接口 IUnknown,这个 IUnknown 接口和 COM 中 IUnknown 接口有同样的作用,如图 4 所示。

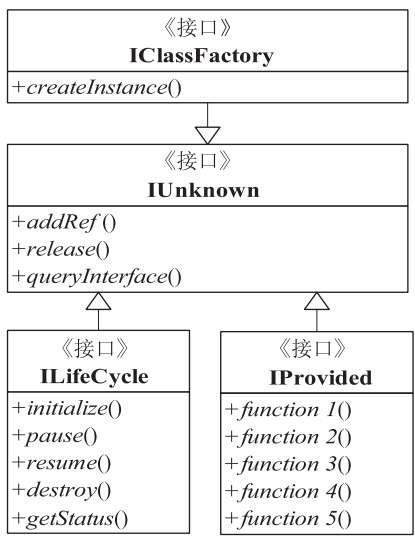


图 4 组件接口

另外,组件必须要实现 IClassFactory 接口和 ILifeCycle 接口,其中 IClassFactory 接口是类工厂,用来创建组件的接口对象。ILifeCycle 接口来实现组件生命周期的管理,组件有以下四种状态:loaded、available、paused 和 destroyed。

当一个组件实例加载到内存中的时候,认为是 loaded 状态;组件在使用前必须通过方法 initialize 进行初始化,初始化进行如内存申请、工程信息加载等预备工作;完成初始化之后的组件处于 available 状态,此时可以为系统中其它组件提供服务;通过调用方法 pause,运行环境可以随时暂停组件服务,组件进入 paused 状态,缓冲的请求会被继续处理,而新的请求会被抛弃;组件最后的状态是 destroyed,在执行 destroy 方法之后,组件进入 destroyed 状态,这时组件已经完成了运行,从内存中清除,释放了占有的资源,不能再被使用。如图 5 所示。

组件除了要实现上述接口外还要实现自己对外提供的服务接口 IProvided。为了保证组件的可重用性,组件的接口一旦确定不再改变。在各个组件

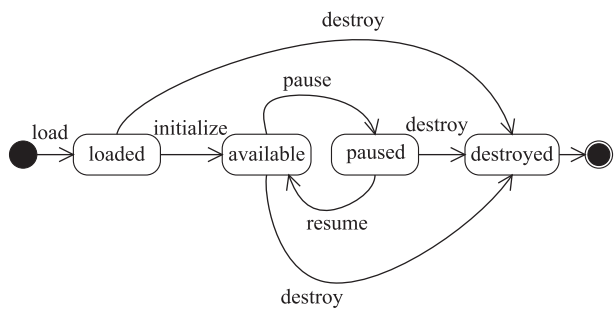


图5 组件的状态图

内部采用单件模式,使各组件在整个系统的运行过程中只存在一个实例。单个组件的升级不影响整个系统的使用。在设计组件时多次采用了桥模式。如在报警组件中,报警类型有 PDR、条件报警等,而报警方式有窗口报警、声音报警等,桥模式的采用使报警类型和报警方式可以动态的组合,减少了组件的代码,增加了组件的可扩展性。

3.3 组件的组装

组件通过复合来建立系统。在复合中,两个或多个组件集成到一起以建立一个更大的实体,而它可能是一个新组件或整个系统,如图 6 所示。

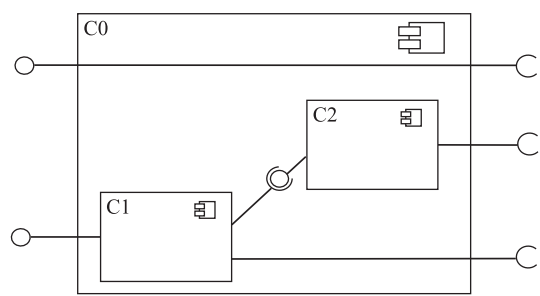


图6 组件的复合

本系统基于 WINCE 系统,采用动态链接库作为组件的实现服务器,组件接口的描述以 C++ 头文件的形式给出,C++ 头文件同时也是对接口的规范。动态链接库并不等于一个组件,同时获取动态链接库和头文件才能使用组件。通过组件接口地址获取组件的服务,从而支持了组件的动态加载和在运行时对接口的迟绑定。

系统启动时,主程序分别获取各组件接口的指针,通过各组件接口的指针调用各自 Initialize 方法

和 Start 方法以开始组件的服务。

系统的运行不需要每个组件都初始化,用户可以根据所需功能的不同定制系统。例如,用户不需要现场监控,则可以在硬件环境中不配备显示设备,系统中不加载界面组件和数据字典组件,这样系统仅不具备界面监控的功能,其他功能如报警、数据发布等都能正常工作。根据硬件资源的不同配置不同的系统,最大限度地节约了成本。

3.4 嵌入式监控系统对已有监控系统的改进

可维护性方面:已有系统基于模块化设计,各模块之间联系紧密不可分割,维护成本高。嵌入式监控系统按功能划分组件,使各组件功能明确、职责单一,降低了系统各模块之间的耦合性,使系统易于维护和扩展。

资源消耗方面:已有系统实时数据库内存冗余大,而且各模块必须同时运行,对处理器和内存都有很高的要求。嵌入式监控系统是可配置的,实时数据库采用树形结构,保证了实时性的同时大幅降低了内存开销。

硬件方面:嵌入式工控机的成本低于普通工控机,体积也小于普通工控机,体积的减小使嵌入式监控系统适应更多的现场监控环境。

本文实现的嵌入式电力监控系统满足了应用系统需求的功能,图 7 是系统实际运行界面的截图。

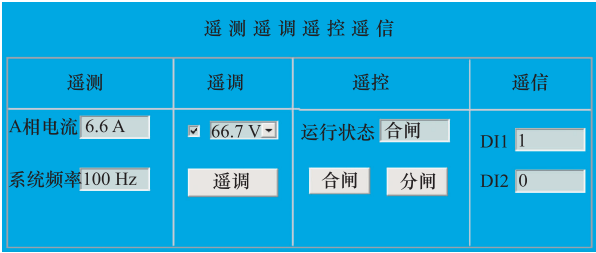


图7 嵌入式电力监控系统的一个监控界面

4 结论

本文在嵌入式平台的基础上,运用组件技术,设计并实现了电力监控系统。本系统以可重用的软件组件为组装模块,支持组装式软件重用,提高了软件生产效率 and 产品质量、缩短了产品交付时

间。用户可以根据不同的需求动态地组装一个系统,增加了系统的可重用性、可配置性和可扩展性。新的系统使整个监控环境体积变小了,成本降低了,可靠性增加了。

参 考 文 献

1 於志文,周兴社,李志刚,等. 基于嵌入软构件的开发技术研究. 计算机应用研究,2003;20(4):12—14

2 房红征,赵贵根,柳克俊. 嵌入式组件模型研究. 微计算机应用,

2005;26(5):521—523

3 Ommering R, Linden F, Kramer J,*et al.* The Koala component model for consumer electronics software IEEE Computer, 2000;33(3):78—85

4 张莉,周兵,柳松. 嵌入式视频监控组件的设计与实现. 微计算机信息,2007;23(35):83—84

5 Hansson H, Åkerholm M, Crnkovic I, *et al.*, SaveCCM-a component model for safety-critical real-time systems. Proceedings of 30th Euromicro Conference, 2004,2004:627—635

Design and Implementation of Component-based
Embedded Power Monitoring System

FANG Hai, WU Jian, ZHANG Yun

(College of Computer Science, Northwestern Polytechnical University, Xi'an 710072, P. R. China)

[Abstract] Through analyzing the characteristics of embedded components and feature of currently famous component models for the embedded system, a component – based method for design Embedded Power Monitoring System was suggested. The use of component software technology improves quality and efficiency of embedded development, and the system is extendable and reusable.

[Key words] component power monitoring system configuration embedded software