

嵌入式应用中运行支撑框架的构件化技术研究^{*}

冯庆桑楠熊光泽

(电子科技大学计算机科学与工程学院 成都610054)

摘要 为增强嵌入式软件的复用性、可移植性,缩短开发周期,通过研究现有的主流构件运行支撑技术,提出了一种面向嵌入式领域的运行支撑体系结构。为实现灵活、开放的嵌入式应用平台,本文首先建立了构件化的嵌入式运行支撑框架模型,然后结合 PDA 手机领域重点分析了开放式 HLA 结构和虚拟 OS 平台等关键技术,最后利用伪代码示例探讨了该领域中应用平台的定制与扩展方法。

关键词 嵌入式构件,中间件,运行支撑框架,构件接口,嵌入式操作系统

Research on Component-Based Runtime-Supporting Framework of Embedded Systems

FENG Qing SANG Nan XIONG Guang-Ze

(Computer Science and Engineering College, University of Electronic Science and Technology, Chengdu 610054)

Abstract To enhance the embedded software reuse and portability, shorten the developing cycle, based on a research for some main popular component-based runtime-supporting techniques, brought forward a kind of runtime-supporting frameworks for embedded systems. In this paper, to realize a flexible and open embedded application platform, a component-based runtime-supporting framework model is firstly constructed. Then, some critical techniques, such as an open HLA infrastructure and a virtual OS platform, are analyzed for the PDA software application platforms. Finally, some methods of customizing and extending those platforms are discussed in the form of pseudo code.

Keywords Embedded component, Middleware, Runtime-supporting framework, Component Interface, Embedded operating system

1 引言

自20世纪60年代以来,嵌入式软件在国防建设和国民经济中得到了日益普及的应用。从航天飞机上的雷达监控系统到 PDA 手机软件,再到日常生活中的家电控制软件,嵌入式技术一直都代表着当代计算机技术的发展方向。但伴随着3G通信时代的到来,分布式系统的发展,计算机操作系统和硬件配置不断提高,人们对嵌入式软件的开发规模、周期、性能、质量等方面也相应地提出了更高、更苛刻的要求,这些都给嵌入式软件的开发带来了新的挑战:(1)如何根据不同需求迅速地升级软件,开发新的业务和功能;(2)如何在异构环境下增强应用程序之间的移植能力和交互能力。基于构件复用和框架复用的软件复用思想是解决上述问题的主要途径之一。

构件复用已经在学术界和产业界得到了广泛深入的实践,其关键是基于中间件技术的运行支撑框架。中间件技术已经成为了当今软件界研究的热点,国内外已经推出了不少的标准和实用产品,其中面向对象的中间件技术和工具发展尤为突出。例如,Microsoft MFC GUI 构架和 OCX 组件是 PC 平台上用于创建图形商业应用的标准;对象管理组(Object Management Group,OMG)针对可移植、分布式应用领域提供了一个完整的体系构架,制定了一套关键规范——对象管理体系(Object Management Architecture,OMA)和它的核心(CORBA 规范)^[1,2];美国 Argonne 国家实验室开发出一套针对实时领域的中间件标准 TAO (Toolkit for Advanced Opti-

mization)^[3];国内东方通公司的消息中间件、交易中间件、对象中间件、应用服务器中间件、数据访问中间件等等。

可见,大多数中间件能很好地解决开发分布式系统时遇到的问题,但却难以有效地支持嵌入式应用平台的抽象和支撑机制的封装,这主要是因为嵌入式软件的开发有其自身的特点:一方面,目前流行的嵌入式操作系统繁多,不同的操作系统和应用平台提供的编程接口不统一;另一方面,嵌入式软件面向过程的开发模式由于其开发周期长、可复用性差、维护成本高等缺陷,越来越不适应嵌入式软件发展的需要。因此,提高嵌入式软件的复用性和灵活性、增强可移植性、缩短开发周期已迫在眉睫,而如何利用面向对象的中间件技术有效地支持嵌入式领域的封装就成为了解决以上问题的关键。本文我们提出了嵌入式构件运行支撑框架的概念和体系结构模型,对实现中一些关键技术进行分析探讨,最后得出结论。

2 体系结构

2.1 基本概念

框架^[1],从面向对象设计角度看,是一个可重用的体系,它将自身划分为一系列既相互合作又相对独立的类,这些类用来产生新的用户应用。一个框架通常规定了应用的基本设计结构,以便将这些应用集成到该框架中。框架的设计与实现大量地使用了中间件技术。

所谓中间件,就是位于平台(硬件和操作系统)和应用之间的通用服务,这些服务具有标准的程序接口和协议,针对不

^{*} 本课题得到国家“八六三”高技术研究发展计划“面向 PDA 手机的嵌入式软件开发平台”(No. 2003AA1Z2210)资助。冯庆 硕士研究生,主要研究方向:嵌入式操作系统、中间件技术。桑楠 博士,副教授,主要研究方向:实时计算机系统、软件工程。

同的操作系统和硬件平台,它们可以有符合接口和协议规范的多重实现。本文提出的嵌入式构件运行支撑框架正是利用中间件技术构造的一种高层体系结构(HLA:High Level Architecture)^[4]。从抽象角度看,HLA 主要由接口规范、设计规则、构件模型和支撑层四部分组成。接口规范屏蔽了底层实现细节,为软件应用平台提供了一致的可编程接口,它主要分为操作系统适配层和应用层两大类。设计规则描述了构件和支撑层之间的关系框架,构件间的信息交互与通信就以它为基础。构件模型定义了开发或扩展构件功能时使用的通用方法,它为应用平台的扩展提供了统一的模式。支撑层包括了一部分接口规范和具体实现,主要负责构件间的信息交互,它屏蔽了构件间通信的差异性,增强了在分布式环境下应用程序的移植能力。可见,嵌入式构件运行支撑框架的应用,将为嵌入式软件平台提供丰富的构件、灵活的定制能力和简单的编程界面。

2.2 体系结构

本文提出的嵌入式构件运行支撑框架,如图1所示,主要采用面向对象、接口封装、动态链接等技术为应用构建了一个一致的、C/S 模式下的嵌入式软件开发平台。该框架利用集成的抽象类将平台内部的系统功能和应用功能封装在各个相对独立的模块中,用户可以根据自己的需求方便地定制和扩展这些功能模块以及选择不同的操作系统。按照功能职责,图中将该构架划分为三个子系统:构件(Component Cluster)子系统、RTI(run-time infrastructure)子系统和 OS 适配(OS Adaptor)子系统:

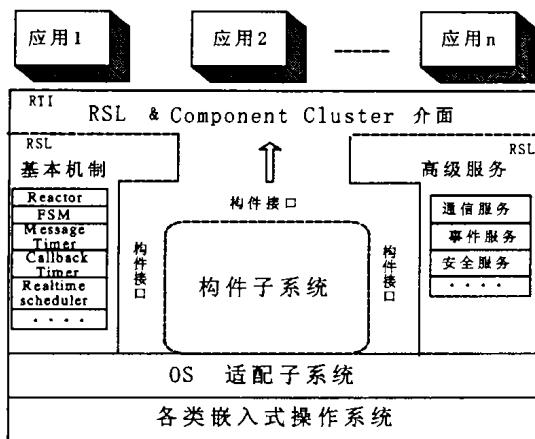


图1 嵌入式构件化运行支撑体系结构

(1)构件子系统 包括面向领域的应用构件库和构件库接口两部分。应用构件库不仅是简单构件和复合构件的集合,更是构件运行时所处的空间,负责对象的适配、创建、装载、释放等生命周期的管理和运行时构件所需的上下文管理(如运行时资源分配、设备配置管理等)。构件库接口是用户和应用构件间互操作的通道,借鉴ACE(Adaptive Communication Environment)^[8]的设计思想,基于规范化的构件模型,我们用抽象基类提出接口,用户无需知道接口实现细节便可将构件组装或扩展,有效地支持了构件对象的运行级组装和即插即用(plug and play)。

(2)RTI子系统 即运行时基础设施,分为运行支撑层RSL(Run-time Supporting Layer)和RSL用户界面,界面是接口规范的表现形式,RSL则是接口规范的具体实现。为适应分布式通信环境,该系统将方法引发和方法执行解耦,利用规范化接口封装方法的实现细节,采用反应式对象(Reactor

Object)+状态机(FSM)机制实现多任务间的通信,以及用定时器(Timer)+实时调度器(Real-time Scheduler)机制满足软实时性需求,并利用动态增加、替换、删除等功能提供各种服务(如通信、安全等),实现了应用间透明的互操作和可移植性,使应用程序易于理解和维护。新增服务须向RSL申请注册。

(3)OS适配子系统 基于嵌入式操作系统多样化的特点,该系统负责隔离下层不同的操作系统,给程序员提供一个与下层具体操作无关的统一的接口,类POSIX接口规范是一个比较好的参考。该系统采用C++的外观包装(Wrapper Facade)模式实现一个最小的操作系统闭集——中断管理、内存管理、定时器、进程管理、设备管理,信号量,然后在实际应用中作相应扩展。大量地使用C++内联可消除额外的方法调用开销,对于性能要求很高的包装,应避免使用虚函数。

利用中间件技术构建嵌入式构件运行框架具有众多优点:①透明的资源管理与服务;②根据需求选择不同的资源管理机制;③减小开发者编程复杂度,但中间件运行时的开销和C++代码运行效率的低下(相对与C代码)对具有实时性需求的嵌入式领域(特别是流媒体技术的出现以及在掌上设备中的应用)来讲,是一个至关重要的瓶颈。因此中间件的实时性问题仍是当今业界研究的热点。本框架在OS上扩展SIM-SCHEDULE^[2]算法,将构件中的方法赋予优先级,采用基于方法连接的可抢占优先级调度机制^[7],能有效地支持嵌入式软件的软实时特性。

3 若干实现技术的考虑

本嵌入式构件运行支撑框架以863项目《面向信息与智能设备的嵌入式软件平台》为需求背景,具有以下特点:

- 规范化。包括构件模型规范化、构件框架规范化、用户编程规范化;
- 软件平台可配置。目标机采用嵌入式Linux,配置项包括系统构件、手机基本功能构件、PDA基本功能构件、应用扩展构件、各种服务等;
- 有效地支持多媒体应用(e.g. mp3、MPEG-4)的软实时特性;
- 可移植性。应用程序可在不同的嵌入式系统间移植;
- 方便的软件开发过程。软件无需修改、重编译、重链接、或频繁的重起运行中的应用,就可被更新和扩展;
- 支持互操作框架。

3.1 虚拟嵌入式OS平台

嵌入式实时操作系统的多样性,要求PDA、手机等嵌入式软件能运行在不同的操作系统之上,用户也可以根据需求特点定制和扩展系统功能。因此,为屏蔽底层各类实时操作系统的差异,向上层软件提供一个统一的接口,通过分析Vx-works、Linux、CRTOS等多种操作系统的共性以及ACE中的外观包装技术,该框架抽象出了一个虚拟的嵌入式操作系统平台。

虚拟操作系统平台利用C++的特性(比如模板、继承、动态绑定),对操作系统进行OO包装,向上层提供类POSIX的、类型安全、健壮(强类型)的C++接口,并且采用多种技术来降低和消除额外的性能开销。在这种结构下,用户可以选择地继承、聚合、实例化各种系统构件。如图2所示,虚拟操作系统平台可分为三层:

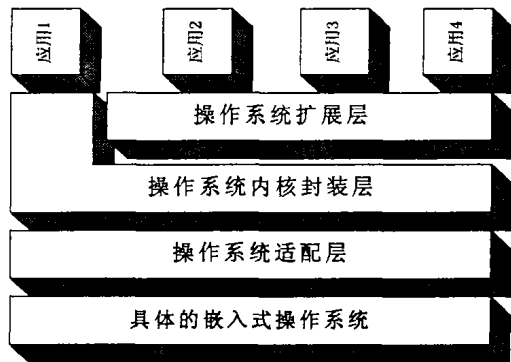


图2 虚拟嵌入式 OS 平台结构

1)操作系统内核封装层 一般而言,嵌入式操作系统具备的主要功能有任务/进程调度与管理、同步与互斥、通信、内存管理、中断管理、实时时钟、I/O 管理等。参照微内核的设计思想,对这些功能接口的实现,首先提取一个最小的操作原语闭集,对其进行类 POSIX 的 OO 封装。该层主要考虑 C 库以外的基本功能接口实现,如线程/进程、信号量、计时器、消息队列等。这些功能接口以抽象类组的形式表现,它们是实现某一具体操作系统主要功能的基类,操作系统扩展层均继承它们而来。

2)操作系统扩展层 考虑到各类操作系统的差异性,该层继承内核封装层,按照并集原则 $\{VOS \text{ 功能集}\} = \{Vx\text{-works 功能集}\} \cup \{\text{Linux 功能集}\} \cup \{\text{CRTOS 功能集}\} \cup \dots$,以接口的形式覆盖各类操作系统的功能。该方法虽然使得虚拟操作系统平台功能损失较小,但是会产生大量冗余,接口参数复杂。用户可以根据需求,考虑代价因素,进行筛选,实现平台的可伸缩性。

3)操作系统适配层 主要负责将平台对外提供的标准功能接口直接或间接的路由到下层具体 OS 提供的相应接口调用。参照构件技术的实现方法,用一个抽象类厂类 class VirtualFactory 提供各接口实例化的方法,这个抽象类厂类由具体操作系统的类厂类继承并实现。另外还需提供一个全局适配函数 AdaptiveOs(VirtualFactory*)来实例化具体操作系统的类厂类,该函数调用成功,则路由成功。

3.2 开放式 HLA

HLA 是用于构建应用平台的通用技术框架,开放式的 HLA 提高了应用的互操作性和资源的可重用性。如图3所示, CBRTSF^[7]作为一种开放式的 HLA 结构,松散了应用构件间的相互依赖关系,实现关键技术主要包括以下部分:

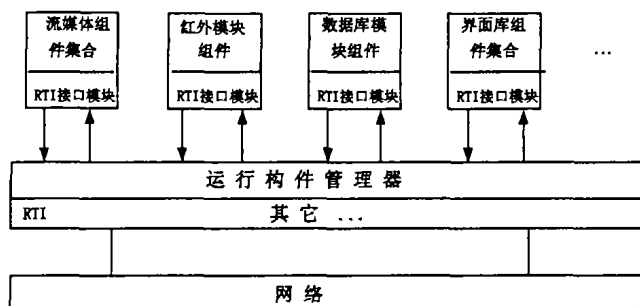


图3 HLA 逻辑结构图

(1)构件接口规范 (a)构件接口:一个应用级构件就是一个接口集,接口又是方法的集合,本设计中,所有构件均是以 Linux 动态连接库(.so)或可执行文件的形式发布的代码

组成的。IComponent-BASE 作为框架基类,是每种构件都必须拥有的基本接口,它提供了最基本功能:① LookupInterface (InterfaceId, * * Iref)使客户在不完全了解构件的情况下获得构件中的其他接口引用;② DuplicateObject ()和 Release Object ()用来控制构件的生命周期;③Component Initialize ()是构件的入口方法,CBRTSF 运行环境调用该方法启动构件工作。IComponent-ADDITON 作为框架类,是每个应用构件拥有的基本接口,它继承 IComponent-BASE 接口,用来配置构件的领域特性。其中 GetPriority (InterfaceID. Method,Priority)和 SetPriority (InterfaceID. Method)表示设置和取得构件接口中的方法优先级。Icomponent- BASE 是所有构件的唯一代表,通过它就可以得到构件的其它接口引用。

(b)包容与聚合:包容是指外部构件包含指向内部构件接口的指针,其主要用途是通过给已有接口加上一些代码以扩展此接口,但编写那些转发调用请求的代码可能是一项极为繁重的工作。聚合是指外部构件直接将内部构件的接口指针返回给客户,它无需重新实现并转发接口中的所有函数,恰好弥补了包容的不足,两者形成互补关系。综合这两种思想,CBRTSF 接口规范规定:接口 Lookup Interface (InterfaceId, * * Iref)负责实现构件内/构件间不同接口的聚合,所有应用级构件须包含一个共有的数据成员:AggregateComponent,它仍以构件形式存在,主要负责完成不同构件接口之间的动态迁移,除此之外,其余接口仍依据包容思想进行封装。如图4所示,聚合原则如下:

```
if (IX,IY∈CA)then call (IX. LookupInterface ( InterfaceId, * * Iref))else realize-link (IX. LookupInterface ()→AggregateComponent. LookupInterface ())
```

说明:聚合接口 IY 时,如果接口 IX,IY 均属于同一构件,直接调用 IX. LookupInterface (),否则要实现“IX. LookupInterface ()→ AggregateComponent. LookupInterface ()”两个方法的连接。

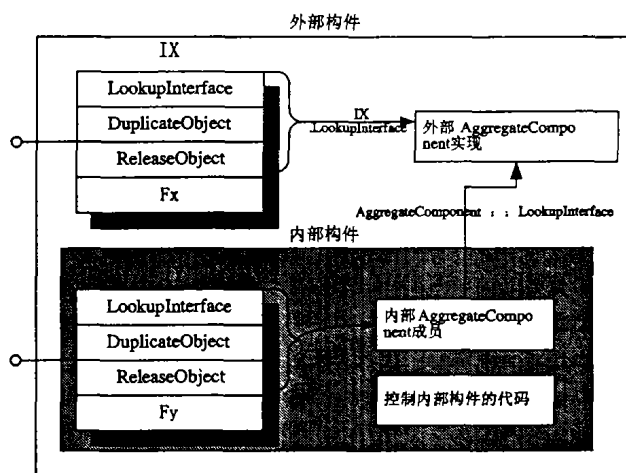


图4 IX 接口聚合 IY 接口结构图

(2)构件运行管理器 作为 RTI 的基本组成部分,构件运行管理器是一个 C++库,为构件的正常运行提供支持,提供对用户及构件都非常有用的构件管理服务,如对象管理、通信管理、实时管理等等。运行管理器的基本功能有:加载(或卸载)并实例化程序中的所有构件(即得到相应构件的 IComponent-BASE 的接口引用),初始化/释放运行环境 Connector 和方法连接控制器 AppLogical^[7],在 Connector 上插入/拔下软件功能模块,在运行中控制和协调各模块之间的通信与交

互等。构件运行管理器是 RTI 的骨架,在多任务环境中的 C/S 开发环境下,按照并发逻辑和应用逻辑分开原则,它采用主动对象模式设计^[6],有自己的接口函数来明确对外展现的功能。任务运行时的互操作过程囊括了客户、RT 库、SO、AggregateComponent、构件五个结构化元素,图5就利用构件创建的示例过程展示了这种交互,过程大致描述为:客户通过调用 RT 库提供的 CreateInstance 接口来启动构件的创建过程;CreateInstance 调用 SO 动态库 SoCreateInstance 接口创建 AggregateComponent;AggregateComponent 再调用 LookupInterface 适配到具体构件实例。该结构支持开放式的互操作框架^[5]。

可见,CBRTSF 把对象对外的接口声明与对象内部的接口实现相分离,一个构件在修改或扩展自己的接口时,原来与它相交互的构件不受任何影响。满足 CBRTSF 模型规范的构件马上可嵌入运行,做到即插即用。因此,构件化的软件框架具有面向对象框架的所有优点,并且可以做到二进制级别上的重用。

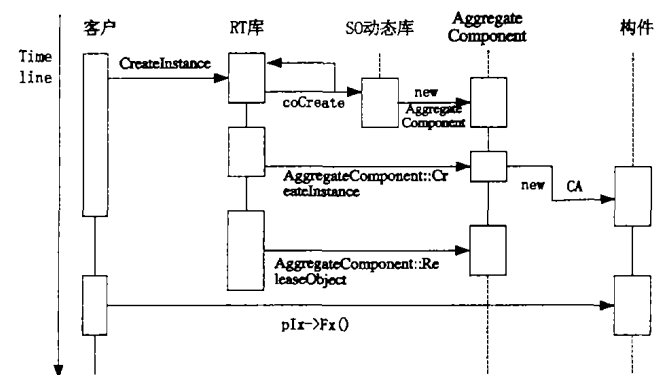


图5 运行管理器构件创建过程示例

术,抽象出面向嵌入式领域的运行支撑框架模型,并且结合 863项目《面向 PDA 手机的嵌入式软件开发平台》探讨了实现过程中的关键技术和伪代码示例。操作系统 SIM-SCHEDULE 算法的扩展有效地支持了 PDA 流媒体的软实时特性,开放式 HLA 和虚拟 OS 平台的实现大大简化了嵌入式软件的开发,增强了软件的可移植性、重用性和可扩展性。

本文的研究仅仅讨论了本地应用,但是只要 RTI 和构件子系统具有分布处理能力,该框架也非常容易扩展到分布式应用。所以下一步的研究主要集中在赋予运行框架的分布式能力,软实时调度框架也是研究的重点。

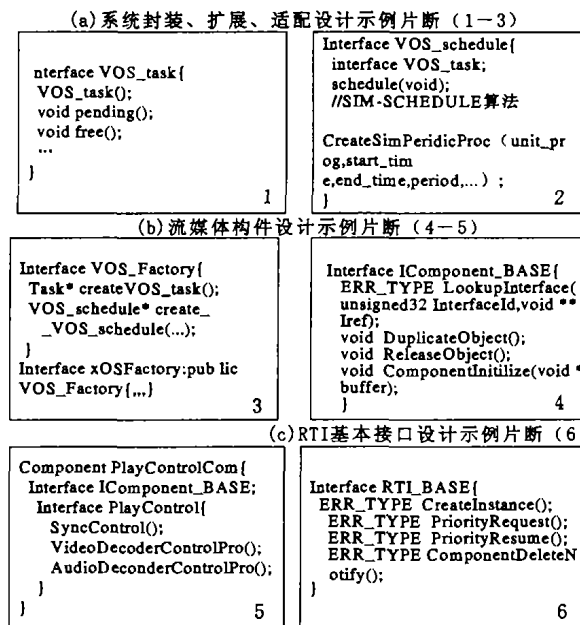


图6 实例设计

4 设计实例

在提供了基础框架和基本库的前提下,本节将以嵌入式 Linux 为具体操作系统,针对 PDA 手机领域探讨嵌入式构件的封装和扩展方法。

PDA 手机是一种具有较高硬件配置的高档手机,在高速移动通信网络的支持下,它可以实现多媒体功能。在本框架中集成多媒体应用构件的过程分为三步,其模型采用伪 IDL 描述如图6所示。

(1)OS 调度模块的封装、扩展和适配。OS 扩展了 RM、EDF 等基本算法,提供 SIM-SCHEDULE 调度算法,有效地支持了多媒体应用的软实时特性,如图6(a);

(2)应用构件封装。按照框架接口规范,图6(b)展示了多媒体应用构件接口的封装过程;

(3)RTI 接口模块。它封装了一系列为应用提供服务的接口,如图6(c)所示。

结论与下一步工作 利用面向对象的中间件技术为面向特定领域的嵌入式软件搭建运行环境是一种新颖的开发思路和设计理念,其中支撑框架的开放性和灵活性是下一代中间件必须具备的特性。本文通过研究现有主流构件运行支撑技

参考文献

- Li Fangxing, Broadwater R P. Software Framework Concepts for Power Distribution System Analysis. IEEE Trans. on Power Systems, 2004, 19(2)
- Fan Chanpeng. Realizing a Soft Real-Time Framework for Supporting Distributed Multimedia Applications. German National Research Center for Information Technology, 1995
- Shenoy P, Hasan S. Middleware versus Native OS Support: Architectural Considerations for Supporting Multimedia Applications. In: Proc. of the Eighth IEEE Real-time and Embedded Technology and Applications Symposium, 2002
- 卿杜政, 李伯虎. HLA 运行支撑框架的研究与开发. 系统仿真学报, 2000, 12(9)
- 黄昱, 王千祥. PKUAS: 一种面向领域的构件运行支撑平台. 电子学报, 2002, 30(12)
- 何先波. 嵌入式操作系统封装层的设计与实现. 计算机应用, 2003, 23(5): 173~178
- 古幼鹏. 一种基于构件的嵌入式实时软件框架. 计算机科学, 2004, 31(12)
- 肖芳雄. 面向对象和模式再嵌入式软件中的应用. 计算机工程, 2003