

组件技术在嵌入式系统中的应用^①

何鹏飞^{1,2}, 何平^{1,2}, 张松阳^{1,2}, 赵鸿博^{1,2}

¹(中国科学院 沈阳计算技术研究所有限公司, 沈阳 110004)

²(中国科学院 沈阳计算技术研究所, 沈阳 110004)

摘要: 研究了提出了通用组件模型不能嵌入式系统中应用的本质原因, 对嵌入式组件模型进行分类以及层次化分析. 并对于嵌入式系统的非功能属性进行分析, 总结了常见嵌入式组件模型对这些非功能属性的支持情况. 然后介绍了组件库的设计, 以及组件模型的未来研究方向.

关键词: 组件技术; 非功能性属性; 分层结构

Application of Component Technology in Embedded System

HE Peng-Fei^{1,2}, HE Ping^{1,2}, ZHANG Song-Yang^{1,2}, ZHAO Hong-Bo^{1,2}

¹(Shenyang Institute of Computing Technology, shenyang 110004, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, shenyang, 110004, China)

Abstract: The essential reason application component model cannot be applied in embedded system is proposed. This paper analysis the hierarchical classification, embedded component model. It also analysis the non functional properties of embedded systems, and summes up the support of common embedded component model for these non functional attributes. And then we introduce the design of the component library, component model and the direction of future research.

Keywords: component-based model; non-functional attribute; layered-structure

1 概述

随着计算机技术的快速发展, 用户需求也越来越大, 使得软件的开发越来越复杂. 因而出现了软件复用的思想, 从软件模块的划分来提高开发效率到采用面向对象的软件复用方法, 再到基于组件技术的软件开发. 组件技术的应用现在已经非常广泛, Windows 编程中使用的各种控件和对话框, ActiveX 控件和 DirectX 的应用等等. 各大著名的公司也提出了自己的组件技术. 微软的 COM、Sun 公司的 JavaBean、OMG(Object Management Group)制定的应用软件体系结构和对象技术规范——CORBA 组件技术. 这些都属于通用的组件模型. 通用组件技术已经发展的相对比较成熟.

但是在嵌入式系统中, 组件模型的应用还在发展

中. 主要的组件模型有: Philips 公司用于消费电子产品的组件模型 Koala、ABB 公司用于现场设备的组件模型 PECOS、比利时 IWT 协会赞助的 SEESCOA 项目的 CCOM 模型等等. 当然也包括很多有名的大学、研究机构、学者提出的嵌入式组件模型. 为什么不用相对成熟的通用组件模型在嵌入式系统中呢? 最本质的原因: 嵌入式系统中非功能属性的约束. 嵌入式系统必须满足时间、可靠性以及受到硬件资源等非功能属性的约束, 但是通用组件模型并没有对这些非功能属性的支持.

2 嵌入式组件模型

2.1 组件模型分类

嵌入式组件模型在建模的过程中可以分为三类

① 基金项目: “高档数控机床与基础制造装备”科技重大专项(2012ZX04010-021)

收稿时间: 2013-11-03; 收到修改稿时间: 2013-12-05

型: 源码组件(编译时组件)、二进制代码组件(连接时组件)和可执行代码组件(运行时组件)^[9]。源码组件是在软件开发过程中产生的, 是实现一个或多个类的源代码文件, 用于产生可执行系统。主要有: ABB 公司的 PECOS 组件模型、Philips 公司的 Koala 模型、MaryLand 大学提出的适用于控制系统的 PBO(Port Based Object)模型。二进制代码组件是源码组件经过编译后产生的目标代码文件或静态、动态链接库文件^[9]。最著名的就是微软的 COM 组件模型, 以及科泰世纪公司将其进行裁剪可以用于嵌入式系统中的 ezCOM 组件模型。可执行代码组件是系统执行时使用的组件, 表示在处理机上运行的可执行单元。最典型的以 JVM 为代表的字节代码组件模型, 在不同平台运行时需要再次编译, 但是不用更改源代码。这三种组件模型各有优缺点, 在嵌入式系统应用中应根据具体需要进行选择。

2.2 组件模型的层次结构

嵌入式系统是为应用而设计的专用计算机系统。嵌入式系统的设计包含了从底层硬件电路的设计到硬件驱动模块的开发, 再到操作系统之上功能软件的设计和用户 UI 界面的设计。因此组件技术的思想应该嵌入到整个嵌入式系统设计的各个层次。图 1 即为组件模型的层次结构:

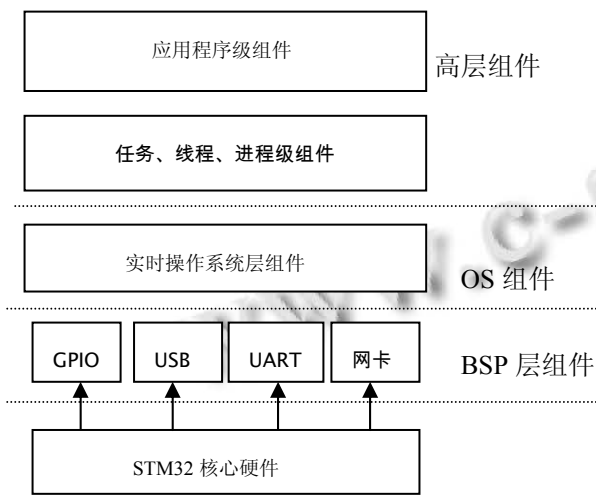


图 1 嵌入式系统组件层次结构

整个嵌入式系统分为 4 或 5 个层次的组件。高层应用软件如果功能比较复杂, 可以在任务之上建立更高层次的应用程序级功能组件。功能比较简单, 我们可以采用任务作为一个独立的组件。对于高层组件,

在满足最主要的非功能性需求的基础上完全可以采用通用组件模型或是经过裁剪后的模型。

加州大学伯克利分校开发的开源操作系统——TinyOS, 是一种专门应用于嵌入式传感器网络的基于组件的操作系统。将嵌入式操作系统进行组件化改装, 必然促进嵌入式系统中组件技术的标准化。因为用户功能软件都是建立在实时操作系统之上的。当然国内也有, 比如上海科泰世纪科技有限公司开发的和欣(Elastos)嵌入式网络操作系统就是一种基于组件的操作系统。并且裁剪通用 COM 组件技术, 实现专用于嵌入式系统的 ezCOM 组件模型。

BSP 硬件相关层, 也是硬件驱动层。对于硬件模块驱动程序的组件化非常重要。不同的硬件处理器, 却经常有一些通用的硬件接口模块。但是因为硬件平台的多样性, 不可能设计出一种组件可以完全不用修改源代码而在各种平台运行。但是我们可以通过编译器让开发人员来手动选择不同的硬件平台和资源, 然后让编译器自动产生硬件相关代码。对于驱动层组件的设计原则首先是使开发人员将驱动组件移植到不同平台时修改源代码量比较少。其次还应保证驱动组件的独立性, 与其他组件之间的低耦合设计。还应保持驱动组件接口的标准化, 可以动态添加不同的接口。

硬件平台层, 我们以 STM32 为例。硬件平台组件化程度比较低。但是在设计硬件电路时, 我们应以组件化的思想来设计, 保证不同硬件模块之间的独立性, 以及模块的“即插即用”性。

2.3 组件模型的结构

理想中的嵌入式组件模型应该使得每个组件能够“即插即用”, 组件内部包含描述一些非功能属性的变量和结构体。组件对外部的接口非常规范、明确, 而且组件容易复用, 组合成复合组件。组件的升级、维护、更新等等都比较容易。但是组件必须有三种基本的结构: 组件体、接口、接口连接器^[6]。

组件体是一个组件的功能实现体, 它的设计是一个组件模型的最重要一点。为了支持平台无关性, 引入组件描述语言 CDL(Component Description Language)对组件进行描述。一个组件至少要包含: 组件 ID、组件描述、输入接口、输出接口、子组件、组件功能实现函数^[6]。当然在嵌入式系统中还要考虑非功能属性的描述。BSP 底层组件只需要在组件体中增加变量或结构体描述时间、占用空间、安全变量等。在

OS(Operation System)组件和高层组件中来通过这些变量计算得到时间、内存消耗、CPU 利用率等属性值。任务级组件可以提供一个统计任务组件，专门用来统计这些非功能属性值。其实在大部分操作系统中都已经增加了统计任务，用来实现这些计算。在一个组件中还可以包含子组件，这样就可以构成复合组件。子组件的接口可以直接作为父组件的一个接口，也可以通过转换提供或者供父组件内部使用。

接口是组件与外部交换信息的窗口，每个组件之间交换信息只能通过接口。引入接口的目的就是为了使“实现与定义分离”。接口用来定义组件的功能，可以使用功能函数指针来表示接口。比如在 Koala 组件模型中，接口就是一组函数指针。在 ABB 公司为现场设备提供的 PECOS(Pervasive Component System)组件模型中的接口又分为输入、输出和双向接口，而且根据触发类型又分为主动型、被动型、事件触发性。接口的类型、触发方式都不一样，因此我们应该引入一种接口的描述语言 IDL(Interface Description Language)来规范接口的描述，而接口的实现则可以采用不同方式。

组件的接口采用不同的数据类型，组件之间就需要采用接口连接桥来进行数据转换。

2.4 组件模型的结构

L.Chung,B.A.Nixon,E.Yu 提出非功能属性包括 12 个属性：可靠性、性能、安全性、效率、费用、数据库、分布性、可用性、耐久性、实施需求、接口需求、物理需求^[5]。而在嵌入式系统中，王刚等人的研究中将其重新分类为五大属性：性能、可维护性、可重用性、可靠性、安全性^[5]。正是由于嵌入式系统中存在这些非功能属性，使得通用的组件模型无法在嵌入式系统中应用，而且也很难形成一个统一、规范的嵌入式组件模型。下面是一些常见的嵌入式组件模型对这些非功能属性的支持情况：

表 1 对于常见的嵌入式组件模型对一些非功能属性的支持情况进行了统计。有通过裁剪通用组件模型使它适用于嵌入式系统的如：MinimumCORBA、ezCOM 等。也有应用于专用领域的嵌入式组件模型。SaveCCM 专用于汽车电子、国防科大的 eBus(基于 MinimumCORBA)用于分布式领域。但是这些组件模型都不能适用于所有的嵌入式系统。

对于非功能属性中的五大属性，性能包括响应时

表 1 组件模型对非功能属性支持

组件模型	非功能属性支持
Koala	没有考虑性能、可靠性等主要非功能属性 ^[6]
PECOS	组件调度执行时间和内存消耗情况
CCOM	采用实时建模，描述时间约束
PBO	基于端口，动态配置系统，满足实时性要
PECT	动态控制开发质量、进度、成本，预测组件属性。从而动态控制软件质量。
SaveCCM	用于车载系统，支持静态配置和构件绑。支持实时性和构件行为的可预测性。
ezCOM	对 COM 进行裁剪，增加了构件的互操作性，构件的简易化包装，构件升级更方便等 ^[7] 。
MinimumCORBA	对通用 CORBA 很多模块进行裁剪，节省资源，增强了互操作性、兼容性、可移植性。

间、延迟时间、吞吐量、并发任务数、资源利用率^[5]。这些参数比较容易描述。在 PECOS 中就描述了调度时间和资源利用率，CCOM 也对时间进行了描述。可维护性描述了对系统的理解、改正、改动、改进的难易程度。基于组件的应用程序开发都具有可维护性比较容易，因为组件是一个自治的实体，理想中的组件甚至可以“即插即用”。可重用性非常重要的一点就是可移植性。通过组件设计的软件系统，可移植性必然能够做到。但是设计组件的过程，必然要接触不同的硬件平台，很难设计出适用于任何平台的通用组件。所以可重用性也是非功能属性中最难以描述的，也是通用组件设计的难点。而可靠性指系统在一定条件下无故障运行的能力。这点在组件的设计中就无法描述，因为它是描述整个系统的无故障运行能力。但是我们可以保证组件的设计最优，从而保证由组件组成的系统无故障运行时间最长。安全性也是现在软件设计中都要考虑的重要一点。组件的设计过程中也应加入安全性检测的代码，但是表 1 中描述的嵌入式组件模型都没有考虑这一点，而且很少有嵌入式组件模型考虑到安全性问题。

在这些非功能属性中最主要的还是性能、可重用性和安全性。这也是我们将来要设计通用的嵌入式组件模型要考虑的重点。

2.5 组件库的管理

当我们的组件设计完成后，如果数量比较多就需要组件库来进行管理。组件的增加、删除、升级、维护、查询、认证等都需要通过组件库来进行统一管理。组件的一些非功能属性也可以通过组件库来实现。比

如组件的安全性,通过组件的认证机制,在组件库中进行存取、查询等都需要经过安全检测,认证通过才能进行操作。

当前一些国内外研究机构对于组件库也提出了很多标准模型。RIG(Reuse Library Interoperability Group)模型是由 RIG 技术委员会提出的一种基于 ALOAF 的三层数据模型:元模型层、模型层、数据层。RIG 主要研究不同组件库之间如何访问的问题。NATO(North Atlantic Treaty Organization)则指出:组件库系统应对每个推荐入库的可复用组件进行定性和定量的评价;在对组件进行配置管理时,应记录下复用者的复用记录,以此改进组件库的查询机制,使用户更好的选取组件。

3 未来研究方向

通用组件模型的研究已经非常成熟了,但是嵌入式组件模型的研究才刚刚起步。主要的模型如表 1 中提出的常见几种嵌入式组件模型。这些组件模型主要可以分为两种类型^[9]: ① 通过裁剪通用组件模型得到的嵌入式组件模型。② 专用于嵌入式系统中的组件模型。正是由于嵌入式系统平台的多样性,很多组件模型是专门用于一些专用领域。如 SaveCCM 是专门为汽车电子产业设计的一种组件模型。ABB 等公司设计的专用于现场设备的 PECOS(Pervasive Component Systems)组件模型。通过裁剪通用组件模型得到的有:科泰世纪公司的 ezCOM、OMG 组织提出的 minimumCORBA 以及基于它的 DeltaCORBA、国防科大的 eBus 等等。这么多的组件模型都不能成为一种嵌入式系统中的通用组件模型。嵌入式通用组件模型的建立对于嵌入式系统中软件的复用,提高软件开发效率非常重要。

通用组件模型的研究主要是针对 BSP 层组件模型,能够设计出一种平台无关性的驱动组件,并且可以通过配置工具来添加删除所需组件。未来嵌入式组件模型的研究主要有以下几个方向^[9]:

- ① 组件模型增加非功能属性分析,这也是解决实现通用嵌入式组件模型的最主要一点。包括非功能属性的描述、量化、评估评测。
- ② 嵌入式组件模型动态配置与静态配置的研究。
- ③ 组件模型的设计时和运行时的组件发布与组合。
- ④ 组件库系统的研究。包括组件库中组件的增删、级维护、查询认证等。
- ⑤ 嵌入式组件模型的安全机制的研究。相信不久的将来,嵌入式组件模型的标准化将会成为现实。这将极大的提高嵌入式软件开发的效率。

参考文献

- 1 van Ommering R, van der Linden F, Kramer J, Magee J. The Koala Component Model for Consumer Electronics Software. IEEE Computure, 2000.5
- 2 Bertolino A, Mirandola R. Modeling and Analysis of Non-functional Properties in Component-based Systems. ELSEVIER.
- 3 Goos G, Bishop J. Component deployment. Proc. of IFIP/ACM Working Conference, CD 2002. Berlin, Germany. 2002
- 4 Wuyts R, Ducasse S. Non-Functional requirements in a component model for embedded systems.
- 5 匡杨.嵌入式构件的非功能属性度量的研究与应用[硕士学位论文].长沙:湖南大学,2012.4.
- 6 马伟民.基于组件技术的人机界面研究[硕士学位论文].杭州:杭州电子科技大学,2009.12.
- 7 朱剑民.ezCOM 构件平台在 Linux 上研究与实现[硕士学位论文].北京:中国科学院研究生院,2003.5.
- 8 蒋银珍.基于硬件构件的底层软件构件开发方法研究.微计算机应用,2005.9,26(5).
- 9 何东海,宋莹.基于嵌入式系统的 minimumCORBA 机制及分析.计算机工程与应用,2012.18.