实时可信服务的构件设计与形式化描述

陈文宇，向 涛，王晓斌，桑 楠，孙世新

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】针对分布式实时嵌入式系统DRES，设计了一种提供自适应服务质量(QoS)保证的构件模型QuOCCM，以形式化方法描述构件，并推导构件各实体之间的自适应过程。QuOCCM由三部分组成，Client、Qoskets和Server，分别以CCM (CORBA Component Model)构件技术实现。Client与应用或功能构件交互，获取当前QoS需求，并触发自适应机制；Qoskets通过对QoS保证框架QuO进行改进，以构件技术实现QoS自适应调整策略；Server提供自适应调整之后的QoS保证机制。研究结果表明，该方法不仅保证了DRES中当前应用环境的QoS自适应需求，而且QoS保证与具体功能应用分离也降低了系统开发的复杂性。形式化方法研究为构件实体间的交互提供了保证。

关 键 词 自适应系统； 构件； 形式化方法； 服务质量

中图分类号 TP302； TP301.2 **文献标识码** A doi:10.3969/j.issn.1001-0548.2011.01.023

Component-Based Software Design and Formal Description in Real-Time Creditable Service

CHEN Wen-yu, XIANG Tao, WANG Xiao-bin, SANG Nan, and SUN Shi-xin

(School of Computer Science and Engineering, University of Electronic Science & Technology of China Chengdu 610054)

Abstract For distributed real time embedded system (DRES), the paper proposes a development model QuOCCM which can provide adaptive quality assurance. QuOCCM is composed of Client, Qoskets, and Server, all of them are implemented by component. The Client interacts with applications, gets QoS requirements, and trigs adaptive mechanism. The Qoskets extends QoS guarantee framework QuO and realizes the system QoS adaptive adjuster using component technology. The Server provides implements of quality assurance. Research shows that the method not only guarantees the adaptive QoS requirement, but also reduces the system complexity by separation of functional path and QoS path.

Key words adaptive systems; components; formal methods; quality of service (QoS)

随着对软件复用及应用研究的深人，可复用的软件构件作为软件复用的核心和基础已越来越得到产业界和学术界的重视。针对基于构件的软件开发技术[1]，研究人员提出了多种构件模型及规范描述构件本质和构件之间的关系，比较有影响的是微软公司的COM/DCOM、SUN公司的JavaBean/EJB以及OMG组织的CORBA。传统的CORBA对象模型有很多不足，如对象功能难以扩展，缺乏统一的软件配置和部署规范等。于是，OMG在1999年将CCM (CORBA component model)纳入CORBA3.0规范的制定中，构件中间件技术通过更高层次的抽象定义，弥补了面向对象中间件技术的一些缺陷，定义了构件的虚拟边界、组件的发布方法以及构件与客户端进行交互的方式。

虽然基于构件的软件开发技术具备提高软件复用率、减少开发时间和开发代价的优点，但是，在DRES中，构件开发技术却面临着诸多挑战，主要表现在：1) 嵌入式应用通常是针对实时、安全敏感的领域，因此系统用于提供QoS保证的一些非功能属性，如实时性、可靠性等必须进行详细设计；2) QoS需求通常不是静态的，应用请求服务级别和系统当前运行状态的自适应过程非常重要；3) 针对嵌入式系统QoS方面的需求，需要改进现有的构件模型，并对其进行规范的形式化描述，从语义上体现QoS方面的需求。文献[2]提出了一种形式化的嵌入式软件源码构件和程序代码设计方法，但该模型对嵌入式系统性能需求设计不明显；文献[3]提出一种针对嵌入式软件的非功能属性的量度模型，但却没有发

收稿日期：2009－07－26； 修回日期：2010－03－09
基金项目：国家863计划(2006AA01Z173，2007AA01Z131)
作者简介：陈文宇 (1968－)，男，博士，副教授，主要从事编译技术、模式识别和形式语言与自动机方面的研究.

挥构件技术在嵌入式开发中的优点。

本文以CCM构件技术为基础，提出实现QoS自适应保证与具体应用或功能构件交互的构件模型QuOCCM，使构件技术在DRES中得到应用，并同时满足系统功能、性能两方面的需求。最后借鉴基于构件刻面的描述方法，对QuOCCM模型进行形式化研究[4]，通过推理验证保证系统设计的正确性。

1　相关工作

1.1　CCM构件模型与实现

CCM构件模型[5]提供称为端口(ports)的多种外部接口，以便与客户、其他构件、CORBA服务等进行交互。构件模型支持5种基本的Ports，如图1所示。其中，刻面(facets)是构件提供的与客户交互的相互独立的一组接口，刻面之间具有导航功能；构件通过接插口(receptacles)持有其依赖的构件或CORBA对象的引用，通过插口构件可以与其他对象进行连接，并调用这些对象的操作，因此刻面和插口定义了构件间同步紧耦合的链接关系；事件源(event source)和事件槽(event sink)定义构件发送和接受事件的能力，从而定义了构件间异步松耦合的链接关系；属性通常用于定义构件在运行时可配置的特性。构件定义用IDL(interface description language)进行描述，所有构件的外部特征都有一个用IDL描述的等价形式，每个构件定义都有一个等价接口。构件的设计者还将为每个构件定义其home，通过工厂模式实现对构件的生命周期管理。

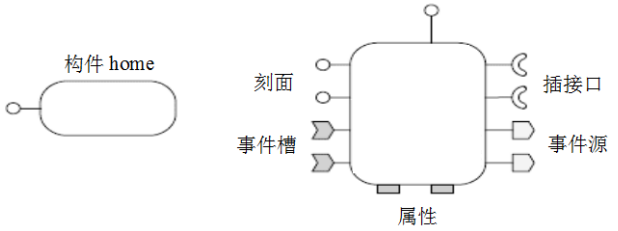

图1　CCM构件模型

构件实现框架定义了构筑构件实现的编程模型。构件实现者利用构件实现定义语言(component implementation definition language，CIDL)描述构件以及构件home的实现。通过编译CIDL描述文件，生成实现框架，自动完成构件的许多基本行为，如导航、身份查询、激活、状态管理、生命周期管理等。开发者对构件框架进行扩展，添加一些用户定制的构件行为，以创建完整的构件实现。

1.2　QuO框架结构

在传统的CORBA应用程序中[6]，客户端通过功能接口实现对远程对象的方法调用。客户端的ORB处理该调用，将该调用发送给目标主机的ORB，远程对象将执行该调用。对于客户端，是功能性的方法调用，如图2示。QuO(quality objects)是BBN公司提出的一个在分布式对象应用中包含QoS自适应保证的框架体系[7]，其支持客户端和服务器端之间的QoS合同、运行期合同监控以及在不同系统环境下的动态适应性调整。QuO应用程序是传统分布式对象计算调用的超集，不仅包含客户端程序、ORB对象，同时也包含实现QoS保证的部分，如合同(contract)、远程对象的本地代理(delegate)、系统条件对象(SysCond)和回调对象(callback)，如图3示。

合同是指定客户期望的服务级别和对象期望提供的服务级别，表示可能被评估的服务级别的操作域，以及当服务质量级别发生变化时的对应措施；每个远程对象的本地代理提供的接口，与远程对象类似，但是增加了本地的基于系统中当前服务质量的自适应行为，服务质量是由合同评估的；系统条件对象提供接口监控系统资源、机制和系统ORB，以供QuO进行合同评估；回调对象提供接口以通知客户或应用程序中的对象。

建立分布式QuO应用程序，除了负责为客户端和分布式对象编写功能应用，还必须开发QoS保证机制，主要包括开发QuO合同、系统条件对象、回调机制以及自适应行为。为了支持QoS保证机制，QuO框架包含服务质量描述语言(quality description languages，QDL)[8]、QuO内核和代码生成器等组件。

QDL主要包括用于指定期望QoS级别的合同描述语言(contract description language，CDL)、用于描述对象和自适应行为的自适应规则语言(adaptation specification language和机制；QuO内核协调对合同的评估以及监视系统条件对象；代码生成器将QDL描述、QuO内核代码以及客户端代码整合，生成一个独立的应用程序。

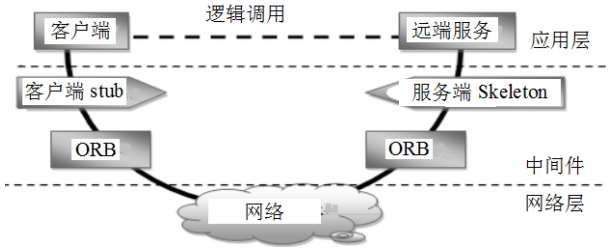

图2　CORBA分布式系统模型

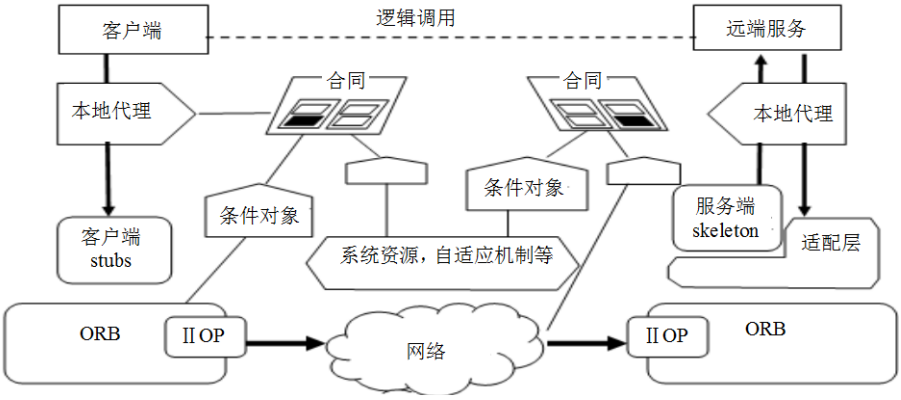

图3 带QoS保证的分布式计算模型

2 DRES的构件设计方法

2.1 网络支撑平台

系统的设计实现基于ACE/TAO技术体系[9]。自适配通信环境(adaptive communication environment，ACE)是可以自由使用、开放源码的面向对象框架，在其中实现了许多用于并发通信软件的核心模式。ACE提供一组丰富的可复用C++Wrapper Facade(包装外观)和框架组件，可跨越多种平台完成通用的通信软件任务。实时CORBA平台TAO是CORBA的一种开源实现，其使用ACE中提供的框架结构对象与模式，实现高效、实时系统的CORBA应用。

2.2 QoS自适应构件模型QuOCCM设计

为了构建可重用性和健壮性更强的应用系统，使QoS确保策略与机制独立于功能构件，通过选择、定制及动态绑定策略与机制为应用系统提QoS保证。本文研究设计了QUOCCM构件模型，其是一种基于TAO的中间件服务，能为应用程序各功能构件提供灵活的QoS确保服务，并且独立于功能构件的实现。为了实现QoS自适应调整，QuOCCM模型必须支持应用开发，包括：

1) 以合同方式定义所期望的服务级别(合同)，即除了规定功能接口，应用还要规定非功能需求，如实时响应、访问控制、同步、带宽管理和独立性。

2) 规定服务级别改变时对应的QoS自适应策略，远端服务的本地代理基于当前操作模式和系统反馈选择不同的功能实现。

3) 衡量、控制系统资源的条件对象和机制。QuOCCM为应用程序提供反馈机制，告知系统当前状态，并且为资源和机制提供标准接口。应用程序可以根据反馈信息触发自适应行为，还可以利用控制接口尽力实现期望的服务。

定义好合同、自适应策略、条件对象和机制后，还必须为QuOCCM定义如图4所示的抽象构件模型[5]，即定义外部接口，以便与客户、其他构件等进行交互，如图4所示。QuOCCM提供QoS定制代理接口、功能代理接口和适配器接口3类接口。QoS定制代理接口可从客户端获取QoS需求，触发QuOCCM内部的QoS自适应调整机制；功能代理接口提供远端功能构件的本地代理，并根据回调对象反馈的自适应策略调用远端服务；适配器接口为应用程序员提供访问QuOCCM内部机制的接口。构件的定义和实现框架通过IDL、CIDL等描述语言编译生成。

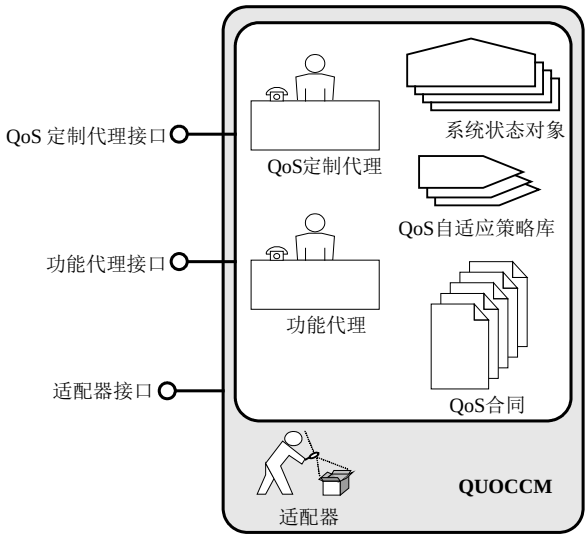

图4 QuOCCM构件模型

2.3 DRES的构件设计与实现

基于构件技术的DRES设计分为功能和性能两部分，其中QuOCCM用独立于功能的构件模块实现系统QoS需求，应用程序/功能构件通过与QuOCCM的交互满足系统功能、性能两方面的需求。QuOCCM工作的原理为：本地代理(client)拦截应用QoS请求，触发自适应机制(Qoskets)，自适应机制通过对系统评估选出合适的QoS保证策略(server)，并通过Qoskets中回调对象反馈给本地代理，最后由本地代

理调用远端QoS保证策略。基于构件的DRES设计框架和详细实现分别如图5和图6所示。

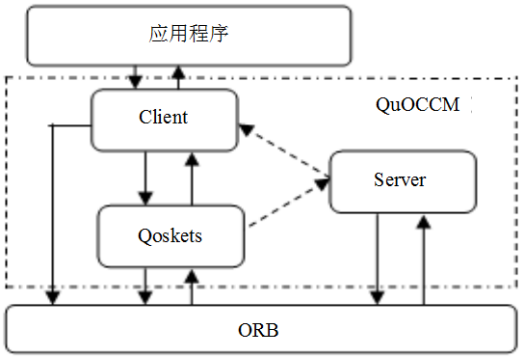

图5　基于构件的DRES开发框架

结合CCM构件技术和带QoS保证的分布式计算模型，可设计出QoS自适应构件QuOCCM，应用程序与QuOCCM交互，共同完成DRES的开发过程。当一个应用/功能构件提出某种QoS请求，Client将拦截该请求，获取QoS参数，QoS定制代理动态绑定其QoS合同及所需系统状态对象，并由Qoskets根据QoS合同和系统当前资源状态，自适应地为该请求分配资源(如网络带宽、执行时间、截止时间等)，并将其反馈到client，最后由client为应用请求指派Server端合适的服务；另外，当系统内部运行状态发生变化而导致系统过载时，也将触发QuOCCM模型的自适应资源调整，以确保任务的QoS，此即为QoS自适应调整的内、外两种触发方式。

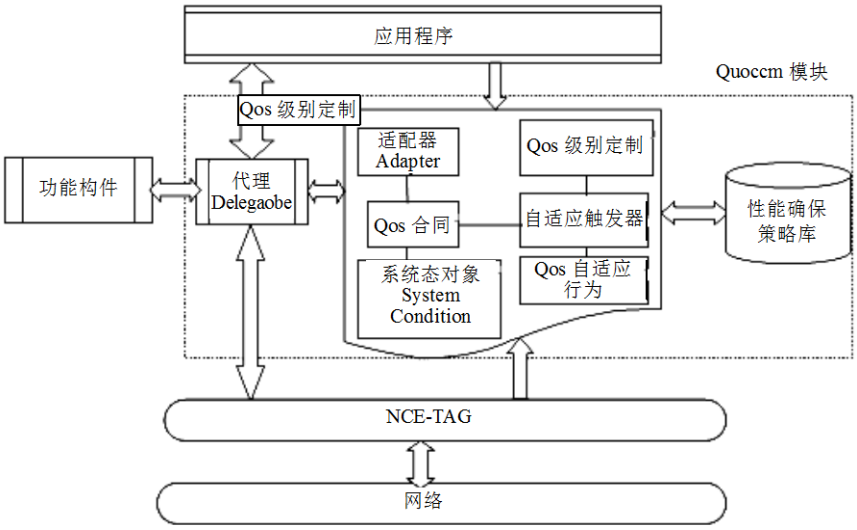

图6　DRES详细实现

3　构件的形式化研究

3.1　构件的形式化描述

在DRES的构件设计框架中，QuOCCM构件为系统提供QoS保证，传统CCM构件提供远端服务，但要更好发挥构件技术的优势，就必须有支持整个软件生命周期并包含大量可用构件的构件库系统，提供有效的构件管理和高效的构件检索，而这些都依赖于准确合理的构件描述模型[10]。QuOCCM构件模型基于传统构件技术，但又增加了许多新的方面，要在DRES的构件开发中充分体现构件技术的优点，就必须提出更准确的构件模型描述。本文借鉴基于构件刻面的描述方法[11]和形式化的BNF方式，从概念、操作规约、接口、实现体、依赖关系、属性等方面刻画构件。

DRES的构件开发框架中，通过不断与QuOCCM交互，传送QoS的相关参数，QuOCCM的自适应调整机制依据传送数据参考合同，动态调用相应的QoS保证策略。于是，QuOCCM被定义为四元组：

QuOCCM=〈BaseInf,Client,Qosket,Server〉

1) BaseInf主要描述构件的基本信息，如编写者、编写语言、运行平台等，且：

BaseInf={〈编写者〉,〈编写语言〉,〈运行平台〉,…}

2) Client主要接收应用的QoS需求，如实时控制中的截止时间、完成时间等，并定义与Qosket交互的接口以及调用返回接口，且：

Client={〈QoS参数〉,〈自适应触发接口〉,〈回调接口〉}QoS参数={〈执行时间〉,〈截止时间〉,〈执行周期〉,…}

3) Qoskets根据QoS参数自适应调用性能保证机制，主要由QoS合同、回调对象、系统条件对象组成；其中，合同包含状态转移和状态域组成，分别描述系统资源发生改变和当前条件下应该采取的自适应行为，通过自适应策略调用具体的QoS保证机

制，且：

Qosket={〈QoS合同〉,〈系统条件对象〉,〈回调对象〉,〈自适应策略〉}

QoS合同={〈状态转移〉,〈状态域〉}

自适应策略={QoS保证的实现体的访问接口}

4) Server是提供各种QoS保证的实现体。

3.2 QuOCCM中实体的适应问题

基于构件的DRES设计中，QuOCCM拦截应用请求，根据其服务需求触发QoS自适应调整(当系统条件对象所监控系统资源改变时也可以触发)，回调对象将自适应策略反馈给本地代理，然后调用远端服务。由此可知，QuOCCM中从client端接收QoS请求，Qoskets根据系统当前状态进行自适应调整，最后调用server端，它们之间是顺序适应[12]，即前者输出是后者输入；前者输入满足用户输入需求，后者输出满足用户输出需求。

结合QuOCCM，下面分别对用户需求和所需构件进行描述：

$$\begin{gathered}\text{Problem}\ (\text{in}\ x:D,\text{out}\ y:R)\\ \text{pre}\quad I(x)\\ \text{post}\quad O(x,y)\\ \text{Client}\ (\text{in}\ x:D_1,\text{out}\ y:R_1)\\ \text{pre}\quad I_1(x)\\ \text{post}\quad O_1(x,y)\\ \text{Qoskets}\ (\text{in}\ x:D_2,\text{out}\ y:R_2)\\ \text{pre}\quad I_2(x)\\ \text{post}\quad O_2(x,y)\\ \text{Server}(\text{in}\ x:D_3,\text{out}\ y:R_3)\\ \text{pre}\quad I_3(x)\\ \text{post}\quad O_3(x,y)\end{gathered}$$

定义 在DRES中，QuOCCM各实体的顺序适应满足以下条件：

1) $:\forall d\in D\,|\,I(d)\Rightarrow I_1(d)$

2) $:\forall d\in D,\exists x\in D_2\,|\,I(d)\wedge O_1(d,x)\Rightarrow I_2(x)$

3) $:\forall d\in D,\exists x\in D_3\,|\,I(d)\wedge O_2(d,x)\Rightarrow I_3(x)$

4) $:\forall d\in D,\exists x\in R_1,\exists y\in R\,|\,I(d)\wedge O_1(d,x)\wedge O_2(x,y)\wedge O_3(d,x)\Rightarrow O(d,y)$

条件1)说明应用请求能使client前置条件成立；条件2)说明client执行后的输出可使Qoskets前置条件成立；条件3)说明Qoskets自适应调整的结果满足Server端QoS实现体的前置条件；条件4)说明问题的前置条件 x 满足Client的前置条件，直到Server执行后得到输出值 y，且该输出值满足用户输出要求，使用户需求描述的后置条件成立。

根据构件形式化描述判断上述4个条件成立，采用顺序适应结构，构件适应过程描述为：

QuOCCM=Client+Qoskets+Server

式中，‘+’表示构件是顺序适应；QuOCCM执行过程是Client接收应用请求；Qoskets再执行自适应机制；Server提供QoS保证。可以证明顺序适应后的QuOCCM的后置条件与用户需求描述的后置条件是等价的，所以验证了QuOCCM实体之间组装的正确性。

4 安全自适应构件开发实例

4.1 总体框架

安全自适应构件组主要由Client构件、Qosket构件、Server构件、接收方构件4个主要构件组成，4个构件分别提供不同的功能。

Client构件主要用于提出各种各样的服务请求，但是其提出的请求主要是基于应用方面的，并不涉及性能保障方面的设计。

Qosket构件主要用于连接Client构件和服务构件，并提供自适应的性能保障策略，在安全自适应模块中主要用于自适应的实现，根据外界的系统资源变化(例如CPU利用率)选择不同的安全算法，既确保信息安全，又确保性能需求。其位于Client构件和Server构件之间，但是对于Client构件和服务构件，又不明确地感知其存在，从而相对透明地实现了性能保障的需求。

Server构件是安全功能的具体实现，将应用端发出的指令信息进行相关处理，主要提供用于实现应用服务请求的各种服务，涉及信息的完整性、信息的保密性和验证发送方身份的算法实现。而且，根据设计的需要，服务构件上也可以存放一些自适应策略中的自适应行为的具体实现。

接收方构件主要用于接收指令信息、验证码、密文等相关信息，对信息进行相应处理，判断信息正确性或发送方身份。Idl定义了各个构件之间的统一接口，用于实现各个构件之间的通信与消息传递。

4.2 安全自适应构件描述及流程

预警机与攻击机通信时，需要安全通信，涉及信息不能被篡改、不能被窃取和确保信息来源正确3个方面。假设预警机为节点*A*，攻击机为节点*B*。本文以预警机与攻击机通信时的安全自适应构件为实例对QuOCCM加以说明。

1) 信息完整新校验功能

*A*节点向*B*节点发送指令信息和该信息的验证

码，验证码通过*A*节点中的安全自适应构件进行实时自适应，该构件根据系统状况(也就是CPU利用率)选择不同的校验算法。*B*节点接收到指令信息，通过攻击机构件对指令信息进行验证，对比接收到的验证码，若相等则认为信息完整；否则，指令信息不完整。

2) 指令信息加密解密功能

*A*节点向*B*节点发送密文，密文通过安全自适应构件进行实时自适应，选择不同的加密算法。*B*节点接收到密文后，与*A*节点共享的密钥对密文进行解密，得到指令信息。

3) 信息发送身份防伪造功能

假设*A*节点是一方的预警机，*B*节点为该方攻击机；*C*节点为敌方干扰机。

A、*B*之间共享防伪密钥，*A*节点利用防伪算法和防伪密钥对指令信息进行处理，然后将指令信息和验证码发送给*B*节点。*B*节点接收到指令信息和验证码，利用相同的密钥对指令信息进行处理，发现验证码正确，则身份确认成功。

*C*节点向*B*节点发送信息，由于没有*A*、*B*节点间共享的密钥，所以发送的验证码是错误的，*B*节点显示身份错误，拒绝执行其指令。

5 总结和展望

QuOCCM以构件技术为DRES开发提供了QoS自适应保证，既保证了功能实现，又满足了QoS需求。QoS自适应构件QuOCCM和应用的分离降低了系统开发的复杂性。

本文借鉴基于构件刻面的描述方法，对构件模型进行形式化描述，并证明了构件实体之间的适应关系，从而验证了基于QuOCCM的DRES开发的合理性。

参 考 文 献

[1] 杨芙清, 梅宏, 李克勤. 软件复用与软件构件技术[J]. 电子学报, 1999, 27(2): 69-75.
YANG Fu-qing, MEI Hong, LI Ke-qin. Software reuse and software component technology[J]. Acta Electronica Sinica, 1999, 27(2): 69-75.

[2] 邓勇, 桑楠, 罗克露. 智能家电嵌入式软件的源码构件设计方法[J]. 计算机工程, 2007, 33(6): 280-283.
DENG Yong, SANG Nan, LUO Ke-lu. Design method of code component for embedded software in smart application[J]. Computer Engineering, 2007, 33(6): 280-283.

[3] 王志刚, 王民北, 骆雷飞. 一个嵌入式软件构件的NFA量化度量模型[J]. 计算机工程, 2006, 32(13): 66-68.
WANG Zhi-gang, WANG Min-bei, LUO Lei-fei. A NFA quantity measurement model for embedded software component[J]. Computer Engineering, 2006, 32(13): 66-68.

[4] 任洪敏, 钱乐秋. 构件组装及其形式化推导研究[J]. 软件学报, 2003, 14(16): 1066-1074.
REN Hong-min, QIAN Le-qiu. Research on component composition and its formal reasoning[J]. Journal of Software, 2003, 14(6):1 066-1074.

[5] 潘慧芳, 周兴社, 於志文. CORBA构件模型综述[J]. 计算机应用研究, 2005, 51(5): 14-17.
PAN Hui-fang, ZHOU Xing-she, YU Zhi-wen. An overview of CORBA componnet mode[J]. Application Research of Computers, 2005, 51(5): 14-17.

[6] 刘军, 王宁生. 基于CORBA的分布式系统的开发[J]. 计算机工程与应用, 2001, 37(15: 22-25.
LIU Jun, WANG Ning-sheng. Developing based-corba distributed application system[J]. Computer Enginerring and Applications, 2001, 37(15): 22-25.

[7] SCHANTS R, MICHAEL J. Packaging quality of service control behaviors for reuse[C]//Proceedings of the Fifth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing. Washington D C: [s.n.], 2002.

[8] LOYALL J P, SCHANTZ R E, ZINCKY J A. Specifying and measuring quality of service in distributed object systems[C]//Proceedings of the First International Symposium on Object-Oriented Real-Time Distributed Computing. Kyoto, Japan: [s.n.], 1998.

[9] 周风余, 宋洪军, 刘涛. 基于中间件技术的异构机器人系统设计及实现[J]. 山东大学学报(工学版), 2007, 37(3): 46-50.
ZHOU Feng-yu, SONG Hong-jun, LIU Tao. Design and realization of a heterogeneous robot system based on middleware[J]. Journal of Shandong University (Engineering Science), 2007, 37(3): 46-50.

[10] 何涛, 钱乐秋, 唐彬. 基于语义检索的构件库系统体系结构[J]. 计算机工程, 2007, 33(17): 10-13.
HE Tao, QIAN Le-qiu, TANG Bin. Component repository architecture based on semantic retrieval[J]. Computer Engineering, 2007, 33(17): 10-13.

[11] 王渊峰, 张涌, 任洪敏. 基于刻面描述的构件检索[J]. 软件学报, 2002, 13(8): 1546-1551.
WANG Yuan-feng, ZHANG Yong, REN Hong-min. Retrieving components based on faceted classification[J]. Journal of Software, 2002, 13(8): 1546-1551.

[12] 陈文, 丁晓明. 构件形式化描述与模糊检索研究[J]. 计算机科学, 2008, 34(9): 292-295.
CHEN Wen, DING Xiao-ming. Research on the formal description and fuzzy retrieving of component[J]. Computer Science, 2008, 34(9): 292-295.

编 辑 税 红