

## 基于COM/DCOM的构件组装工具的设计

蒋 韬, 赵文耘, 张 志

(复旦大学计算机与信息技术系软件工程实验室, 上海 200433)

**摘 要:** 软件构架技术的研究重点在于软件构架描述语言及其支持工具。该文通过对现有构架技术的研究, 提出了基于COM/DCOM标准组件技术来构造构件组装工具的相关技术。基于COM/DCOM构件的软件开发环境的设计, 描述了一个原型系统SOLOSTUDIO。

**关键词:** 软件构架; 构架风格; 构架描述语言; 接口定义语言

## Design of Component Assembling Tool Based on COM/DCOM

JIANG Tao, ZHAO Wenyun, ZHANG Zhi

(Software Engineering Laboratory, Department of Computer and Information Technology, Fudan University, Shanghai 200433)

**【Abstract】** The research emphasis of software architecture is software architecture description language(ADL) and its support tools. Based on the research on current SA-related levels, this paper provides technologies used to construct a COM/DCOM component-based assembling tool. It focuses on design of the environment, and describes a prototype system called SOLOSTUDIO.

**【Key words】** Software architecture; Architecture style; Architecture description language; Interface definition language

在构件组装中, 软件构架的风格是核心问题。最近, 软件开发的重点已经从代码行转移到具有不同粒度的构架元素(比如说构件、连接器等)中去。本文引入一种称之为 C2 (Component and Connector)的构架风格。这种风格一开始被设计成一种支持图形用户接口应用程序的风格, 但是也被实践证明可以支持其他风格的应用程序。一般来说, 一些早期的构架描述语言比如Wright、Rapid、Darwin只提供各自面向专业领域的支持工具, 随后的一些构架描述语言比如Acme和C2, 有着完整的一套与专业领域无关的构架组装工具的支持, 最新的版本分别是Acme1.0 和Archstudio3.0。但是, 这些工具虽然在各种通用组件标准COM/DCOM、EJB以及CORBA和构架领域的整合方面作了一些研究, 但是研究的结果却并不令人满意, 比如由于这些通用组件标准本身的复杂性, ACME以及ARCHSTUDIO并没有很好地对COM/DCOM组件进行包装形成可以被组装的构件, 而这些构件是系统组装所必需的成分, 同时ARCHSTUDIO等工具并不支持和构件库的连接, 所有待组装的构件都是手动编写的代码, 这些都影响了构件组装自动化的程度。本文重点讨论基于COM/DCOM技术的构架组装工具的设计和实现, 讨论将COM/DCOM组件对象作为构架组件的内部对象时和整个系统构架之间的关系, 以及如何从构件库中提取COM/DCOM构件并且自动包装的详细过程, 解决了构件组装从构件库到包装再到组装一系列过程的自动化的问题。

### 1 C2构架风格

C2是一个基于构件和消息的构架风格, 用来支持具有图形用户接口应用程序的需要, 同时也具有支持其他类型应用程序的潜力。一个C2构架可以看成是按照一定规则由连接器连接的许多组件组成的层次网络(如图1), 虚线部分代表其中可能省略了包括许多的组件、连接器以及联结的构架网络。

C2构架风格最重要的特性是“低层独立性”, 它在构件的可替代和可重用性方面具有明显的潜力, 也就是说, 一个构件完全独立于它上层的构件。如果一个构件的“顶”域和在特定的构架中的上层构件的“底”域有密切联系的话,

那么它的可替代性就大打折扣——它只能被具有相似约束“顶”域的构件所代替, 因为这个原因, C2构架引入了“事件转化”的概念, 域翻译器把一个构件的要求转化为接受方能够接受的特定形式, 同时也把通知转化为该构件能够理解的形式。C2设计环境的一个目标就是支持完成这个任务。

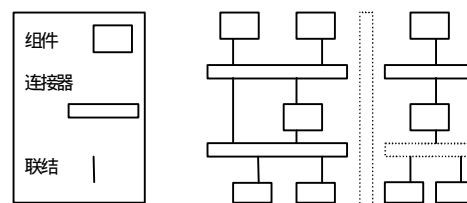


图1 C2构架风格图

### 2 基于COM/DCOM技术的构件组装工具CBACT

本文的主要讨论标准组件对象模型技术COM/DCOM和系统构架之间的关系, 实现了基于COM/DCOM技术的构件组装工具CBACT(Component-Based Architecture Composition Tool), 该工具的主要功能是从构件库中搜索, 下载并且包装所需的COM/DCOM组件, 然后组装成一个应用程序或者一个粒度更大的构件。

#### 2.1 构件组装工具的构架

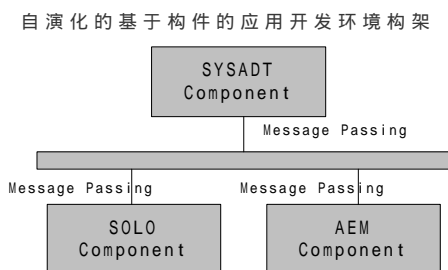
基于COM/DCOM的构件组装工具CBACT采用自演化的方式设计并且实现。即组装工具本身也由构件组装完成, 这就说明工具能够组装更复杂的系统。这样设计有几个好处:

基金项目: 上海市科委科技攻关计划基金资助项目(025115014); 国家“863”计划基金资助项目(2002AA114010); 国家“863”计划基金资助项目(2001AA113070)

作者简介: 蒋 韬(1978-), 男, 硕士生, 主研方向: 软件工程; 赵文耘, 博导; 张 志, 硕士生

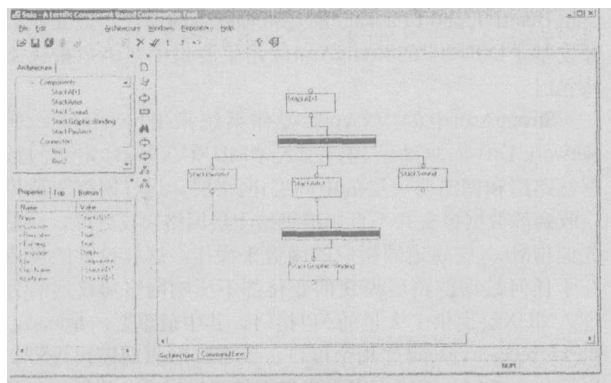
收稿日期: 2003-11-13 E-mail: jiangtao@cn.ibm.com

- 图2是基于COM/DCOM构件的组装工具自身的构架图。



自演化的方式不仅具有很强的灵活性(可以任意修改组装工具的某一个功能而不影响到其他功能),而且具有很高的可扩展性(随着研究的不断深入,可以添加所需的功能而不需要修改整个系统的代码)。但是,引入自演化方式有可能带来新的问题,比如如何保证所有构件的接口一致,保证新加入构件和其他构件完全兼容,保证功能不重复或者重复较少等,这些通过引入依赖检查构件,构架约束检查构件以及类型检查构件得以解决。

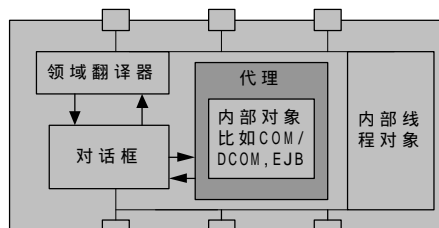
基于COM/DCOM的构架组装工具CBACT如图3所示。



- (1)在组装区可视化地设计系统构架；
- (2)按照剖面信息到构件库查询所需要的构件；
- (3)生成构架描述文档，保存构架描述；
- (4)对每个构件进行包装，生成包装类；
- (5)最后生成应用程序或者子构架。

### 基于C2风格的ARCHSTUDIO组装工具中所有的构件包

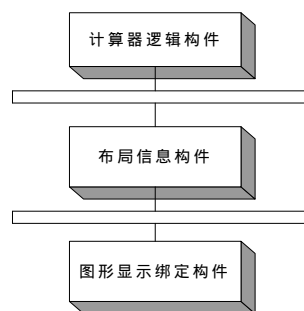
构件的包装是在用户已经完成对构件的获取和设置以后，系统根据该构件的接口描述信息，自动生成一个该构件的包装类。新生成的包装类符合C2构架风格的规范，使构架的一切行为独立于构件的内部对象信息，而具体组装构件的内部构架如图4所示。



构件的包装信息主要是构件的描述信息，需要描述的信息包括：构件常规信息(记录构件的基本信息)，构件片面信息(供构件库检索构件使用的信息)，接口信息(包括接口ID、接口名称，接口方法和事件，这些信息能在可视化的构件组装平台中提供给用户察看，用户可以根据系统需要选择使用的方法和事件)。自动生成的包装类应该包括下面内容：

- (1)接口变量。每个接口都用一个独特的变量来表示，调用构件的方法时，就调用该方法相应的接口变量的方法。
- (2)构件方法。对于构件的每个方法，都生成一个相应的包装类的方法，该方法的名称和参数都和构件的方法相同，该方法的实现就是调用该方法所属接口变量的相应方法。
- (3)端口方法。为构件的每个端口都生成一个方法，该方法有一个参数，就是该端口的接收消息。该方法的实现就是将该端口中的一系列操作转化为可以执行的代码。初始化和消息处理方法，初始化方法主要完成构件的实例化和事件连接处理。

这里通过基于COM/DCOM构件组装工具组装出的一个实例来演示该工具在实际系统组装中的应用，见图5、图6。



(下转第66页)

## 4.2 零拷贝的性能

测试中修改了Intel开发包核心库中StrongArm参考设计,将收到的报文基本不做拷贝,直接转发,同样使用开发板两个100Mbps端口,结果如图2、图3。

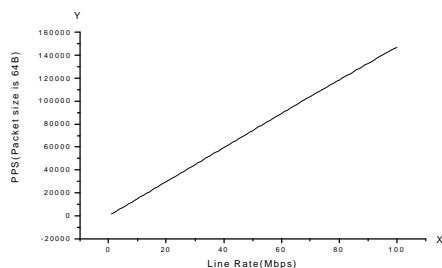


图2 零拷贝PPS曲线

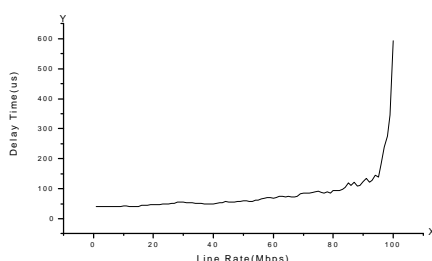


图3 零拷贝时延曲线

从图2、图3可以看出使用零拷贝时网络性能符合预期的效果。但由于StrongArm处理速度和存储资源的限制,在使用100Mbps最大速度发送报文,平均时延立即剧升。值得注意的是,该曲线的测试环境是只使用了IXP1200开发板的两个100Mbps端口作一发一收使用,显然全部使用8个100Mbps端口将使pps降低。

## 4.3 两次拷贝的性能

使用IXP1200开发包中StrongArm原有应用程序进行测试。从图4、图5中可以看到存在两次拷贝情况下网络pps会随着报文发送速度的上升而迅速下降。同时,网络报文平均时延在速度上升不多(11Mbps)情况下就会急剧上升,数值与零拷贝相应发送速度的平均时延形成鲜明对比,表明了改进系统模型的必要性。图中后来之所以又出现时延有所下降是因为大量的报文已被丢弃。

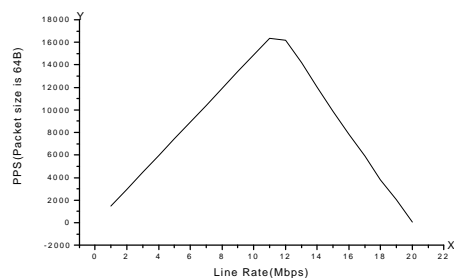


图4 两次拷贝PPS曲线

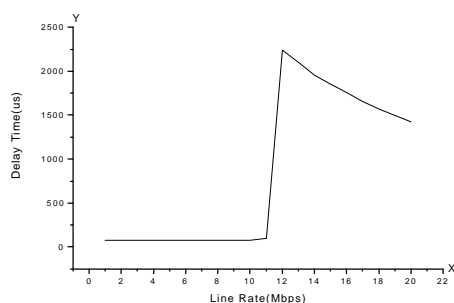


图5 两次拷贝时延曲线

## 5 结束语

我们已经利用改进后的软件模型在IXP1200上实现了具有基本功能的路由器原型。由于使用改进后的软件模型,系统具有很大的灵活性,进一步的性能改进方案和其它应用很容易在这样的模型上展开。例如有MPLS代码则很容易移植到这个平台上来。接下来的研究将是继续研究IXP1200网络处理器性能提高的途径和开发多种网络增值应用。

### 参考文献

- 1 WindRiver Inc. VxWorks Network Protocol Toolkit User Guide 5.4 (Edition 1). 1999-07
- 2 Intel Inc. IXP1200 SW Ref Manual.2001-06
- 3 Intel Inc. IXP1200 HW Ref Manual.2000-12
- 4 卢春鹏. 一种嵌入式系统的内存分配方案.单片机及嵌入式系统应用, 2002, (12)
- 5 谭章熹,林 闯,任丰源等. 网络处理器的分析与研究. 软件学报, 2003,14(2)

(上接第63页)

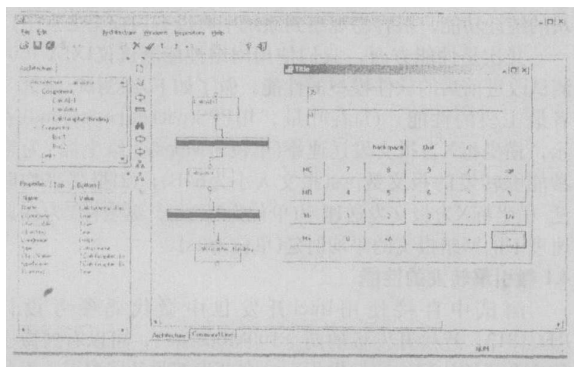


图6 计算器的实例

一个简单的计算器包括3个构件: TCalculatorADT, TCalculatorArtist, TgraphicsBinding。其中所有计算器中需要保留的数据结构和计算逻辑都保存在TCalculatorADT构件之中,但是它并不负责用户界面的显示和用户事件的响应,而这些功能全部放在TGraphicsBinding构件中, TCalculatorArtist构件仅仅是保存用户界面布局信息的构件。

### 参考文献

- 1 杨芙清,梅 宏,李克勤等.支持构件复用的青岛III型系统概述. 计算机科学,1999, 26(5): 50-55
- 2 杨芙清,梅 宏,李克勤. 软件复用与软件构件技术. 电子学报, 1999, 27 (2): 68-75
- 3 Perry D E, Alexander L. Foundations for the Study of Software Architecture. Wolf in ACM SIGSOFT Software Engineering Notes, 1992, 17(4): 40-52