CAR 构件继承机制及其在图形构件中的应用

杨青松 1，陈 榕 2，殷人昆 3，杨维康 2

（1. 清华大学深圳研究生院软件工程中心，深圳 518055；
2. 清华大学信息技术研究院操作系统与中间件技术研究中心，北京 100084；
3. 清华大学计算机科学与技术系，北京 100084）

摘 要：构件继承对于软件复用具有非常重要的意义，为了适应网络发展和 Web 服务应用的推广，科泰世纪科技有限公司提出了 CAR 构件技术。文章首先介绍了软件构件的概念和 CAR 构件的提出，然后对比 COM 和 CORBA 的构件复用方式，重点介绍 CAR 构件的继承机制及其特点，并结合“和欣”图形构件设计的实例，说明该继承机制在图形构件中的实现方法。

关键词：构件；继承；CAR；软件复用

中图分类号：TP311.52

Inheritance Mechanism of CAR Component and Its Application in Graphics Components

YANG Qing-song1，CHEN Rong2，YIN Ren-kun3 ，YANG Wei-kang2

(1.Software Engineering Center of Graduate School at Shenzhen, Tsinghua University, Shenzhen , 518055, China;
2. Operating System and Middleware Technology R&D Center, Research Institute of Information Technology, Tsinghua University, Beijing, 100084, China;
3. Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China)

Abstract: Component inheritance is very significant for software reuse. CAR component technology was created by Koretide Corporation to adapt the development and application of web service. Compared with COM and CORBA, the inheritance mechanism and characters of CAR component are mainly introduced in this paper. The implementation of this mechanism is also discussed, including the design and implementation of graphics components using CAR component inheritance mechanism.

Key words: component; inheritance; CAR; software reuse

1 引 言[1]

1968 年 Mcilroy 在 NATO 软件工程会议上首次提出了软件复用的思想[1]，他倡导软件应像其他机械一样由一些部件（构件）拼装而成。三十多年过去了，软件的复用技术也不断的发展和成熟。面向对象的 C++编程语言的出现是软件复用的一个跨越，类的封装和继承为代码的复用提供了非常有力的保证，当从父类派生出子类时，子类可以继承父类的成员定义和实现。但是这种基于继承机制的类的复用，只是源代码级的重用，在源代码不可得的情况下（构件产业化的发展趋势下，这种情况很普遍），就变得毫无意义了[2]。在面向对象环境下出现的构件技术比传统的代码复用有了很大的提高，为软件的大规模复用提供了可能。构件复用作为一种现实有效的软件开发方法正受到越来越多的重视和研究。

1.1 构件的概念

软件构件（也称软件组件）是软件系统内可标识的、符合某种标准要求的构成成分，类似于传统工业中的零部件。广义上讲，构件可以是需求分析、设计、代码、测试用例、文档或软件开发过程中的其它产品。狭义来说，一般指对外提供一组规约化接口的、符合一定标准的、可替换的软件系统的程序模块。通常情况下是指后者。

软件构件有两个特征：（1）有用性，指构件完成的功能是有用的，也就是其功能可出现在很多应用软件中。（2）易用性，指构件要有很好的包装，能很方便地使用它。一般来讲，构件的包装要符合一定的标准。目前软件构件的标准大致有：微软提出的 COM/COM+、SUN 公司提出的 JavaBean/EJB、OMG 提出的 CORBA、Borland 公司提出的 VCL、微软与 IBM 提出的 WebService 等。[3]

1.2 CAR 构件技术的提出

随着网络技术的发展，Web 服务的概念得到越来越深入的应用。在这种背景下，以构件技术为基础的“软件工厂”式的软件开发以及支持这种构件软件的开发和运行平台的产生就显得尤为必要。科泰世纪科技有限公司的 CAR 技术就是在这种情况下诞生的。

CAR 构件技术是面向构件的编程模型和编程规范，它规定了一组构件间相互调用的标准，使得二进制构件能够自描述，能够在运行时动态链接。CAR 技术在总结面向对象编程、面向构件编程技术的发展历史和经验的基础上，为更好地支持面向以 Web Service（WEB 服务）为代表的下一代网络应用软件开发而发明的。CAR 很大程度地借鉴了 COM 技术，保持了与 COM 的兼容性，同时对 COM 进行了重要的扩展。[4]

基金项目：国家高技术研究发展计划（863 计划）项目（编号 2001AA113400）。

作者简介：杨青松（1978-），男，四川省西充县人，硕士研究生，主要研究方向为计算机应用技术；陈榕，男，清华大学信息技术研究院操作系统与中间件技术研究中心副主任，主要研究方向：网络操作系统，构件技术；殷人昆，男，清华大学计算机系教授，主要研究为软件工程；杨维康，男，清华大学信息技术研究院操作系统与中间件技术研究中心主任，主要研究方向为嵌入式操作系统。

因此，CAR 技术继承了面向对象编程良好的编程思想，并且在构件封装、继承等方面的创新更好的支持了构件软件工厂化生产，同时也有利于构件化编程技术的推广和应用。

2 CAR 的构件继承机制

2.1 CAR 的构件继承机制的特点

微软 COM 的重用性不同于 C++语言，COM 是建立在二进制代码基础上的标准，因而 COM 组件的重用是二进制代码级的。按照 COM 的标准，有两种重用模型用来重用已有 COM 对象的功能：包容和聚合[5]。COM 的包容与聚合在一定程度上实现了软件的复用。但是 COM 的复用存在下面的问题：（1）包容是最简单的构件复用方法，但包容不能重写被包容对象的接口方法实现，只能在外面“包一层”，不能实现复用时的多态性。（2）聚合只是把被聚合对象的接口直接暴露给外面，没有机会改变被聚合对象的接口方法实现，只是让外界看起来像一个大对象。

CORBA(Common Object Request Broker Architecture)是 OMG 在 OMA(Object Management Architecture)基础之上定义的对象请求代理 ORB 的公共结构。CORBA 构件的复用是直接针对目标代码的复用，不必对代码进行改写、编译等工作。与 COM、JavaBean 相比，分布式构件更突出的地方在于构件间的连接，这也是分布式构件复用的主要方式。CORBA 构件之间的连接除了可以采用一般的静态方法建立之外，还可以在运行过程中建立起来，即只有当客户方构件主动向服务器构件发出连接请求时，才建立起连接关系[6]。不难发现，CORBA 构件的复用方式使复用者无法看到被复用对象内部的结构，也无法根据自己的需要修改它们。

C++的虚函数特性实现了面向对象的运行时多态性。运行时，在基类里调用虚函数，会跳到其派生类的方法中执行，派生类可以重载和调用其父类的虚函数。因此，为了解决 COM 和 CORBA 不能重写被复用对象内部的方法的缺点，CAR 在 COM 包容和聚合的基础上，实现了类似 C++的虚函数特性。从而使得基构件调用虚接口方法可以跳到相应派生构件的接口方法中，派生构件可以直接重载和调用继承于基构件的接口方法。

CAR 构件继承机制的特点主要体现在以下几个方面：

1.当基类的一个接口是“虚接口”(接口定义中有 virtual 属性），那么其派生类将可以覆盖该接口的所有方法。这也是 CAR 构件继承机制最重要的特点，区别于 COM 和 CORBA 不能对基构件的接口进行改写的缺点，实现了构件级别的多态性。

2.基类是一个构件，具有构件的特性，包括封装性，基类不须暴露其内部结构。基类只要接口不变，无论怎样升级更新或改变内部结构，派生类都不需要重新编译。因此，CAR 构件的这个特点使得构件复用变得非常容易，复用的水平也大大提高。

3.CAR 构件继承的本质实现是通过特殊接口的调用来实现构件间的迟绑定，完全遵循 COM 规范。CAR 与 COM 的兼容性可以让 CAR 构件得到更大范围的支持和应用。

4.接口定义时支持 private、protected 关键字，类似 C++的相应关键字，用于指示该接口是否提供给外界或派生类使用。

2.2 CAR 构件继承的实现

为了支持面向构件的编程模式，“和欣”构件技术开发了 CDL 语言，CDL 是 Component Definition Language 的简称。一个 CDL 文件用来说明一个 CAR 构件，用于定义构件中的类、接口以及接口方法等信息。每个 CAR 构件都可以由一个或多个类组成，每个类可以提供一个或多个接口，每个接口中可以定义一个或多个方法。

在 CAR 构件的 CDL 文件中声明接口属性的时候可以使用下面的几个关键字：

- **virtual** 有此属性的接口派生构件可以重载和调用此接口的方法，基构件可以调用派生构件中此接口的方法，该接口也可以为构件外部访问。
- **protected** 有此属性的接口只能在派生构件内部使用，构件外部不能访问。并且，如果接口为事件接口，不能添加此属性。
- **private** 有此属性的接口不能被继承，亦不能被构件外部访问。

举例来说，如果需要开发一个构件，实现对车辆的基本操作和控制功能，可以定义一个 Vehicle 类，其 cdl 代码如下：

```
class CVehicle {
    virtual interface IVehicle;
    virtual icallback IVehicleCallback;
protected :
    interface IVehicleHelper;
};
```

构件类 CVehicle 定义了两个 virtual 属性的接口 IVehicle 和 IVehicleCallback，还有一个 protected 属性的接口 IVehicleHelper。因此，在开发 CVehicle 类的派生构件的时候，IVehicle 和 IVehicleCallback 接口能够被派生构件继承和使用，也能够被外部构件访问，而 IVehicleHelper 接口只能在派生构件的内部使用。

图 1 给出了 CAR 构件继承实现的代码结构图，构件 A 实现了接口 IA，其接口属性为 *virtual*。如果构件 B 继承了构件 A，那么通过对构件 B 的 CDL 文件编译可以为构件 B 生成相应 C++的实现代码（这是由

CAR 的自动代码生成工具实现的）。包括构件 B 自己定义的接口方法，以及继承的构件 A 的接口 IA 的实现方法，并创建基构件 A 的对象，实现多态性的接口指针初始化等，这些都是 CAR 构件技术对构件继承在实现上提供的支持。

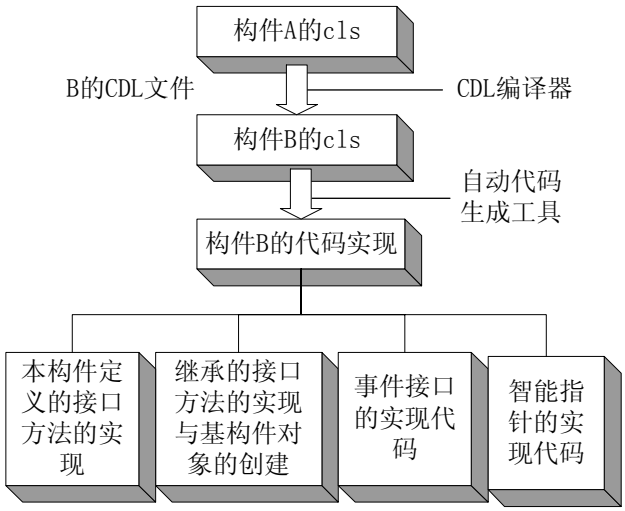

图 1 CAR 构件继承的实现（代码结构）

CAR 构件继承的实现重点需要解决两个问题：一是派生构件能够调用基构件的接口实现；二是在基构件里面调用虚接口的时候，可以跳到派生构件类的方法中执行。下面以图 1 为例具体叙述：

对于第一点，如果派生构件 B 需要调用基构件 A 的接口实现，在派生构件的构件类构造时，可创建构件类 A 的实例，通过类型转化得到，这时需要用到 IARef & MyParentIA() 函数，该函数用来提供指向基构件 A 的接口 IA 的智能指针。

对于第二点，如果基构件 A 想要得到派生构件 B 相应的 virtual 接口指针，就需要使用 IARef & MyVirtualIA () 函数，该函数用于提供指向继承此接口的派生构件 B 中对应的接口智能指针。这个功能是通过一个内部接口 IvirtualInheritHelper 来完成。这个接口只有一个函数 SetMyVirtualInterfacePtr (IUnknown * pMyVirtualInterface, REFIID riidVirtualInterface)，这个函数由继承某 virtual 接口的派生构件的实现内部自动调用。传入派生构件某 virtual 接口指针以及接口 id，SetMyVirtualInterfacePtr 函数内部会 QI 基类的 IVirtualInheritHelper 接口指针，由此接口调用基类的 SetMyVirtualInterfacePtr 函数，在基类的 SetMyVirtualInterfacePtr 函数中会有相应变量接收 pMyVirtualInterface，并由相应的函数返回。从而基构件就拥有了派生构件的 virtual 接口指针了。

3 CAR 构件继承在图形构件中的应用

“和欣”是以 CAR 技术为基础开发的嵌入式操作系统，它为用户提供了一个图形系统构件库。使用图形系统构件库，用户能够更方便的开发自己的图形构件。该构件库由三类构件组成，分别为：GDI 类构件、控件类构件和图像类构件。图形系统构件库由 GDI 构件集和控件集组成。GDI 构件集向外提供图形设备接口，而控件集由控件类构件和图像类构件组成。控件集构建在 GDI 构件集之上，它们的实现使用了 GDI 构件集提供的基本的图形绘制功能。“和欣”图形系统提供的控件集可以使用户能简便、灵活的构造出复杂的图形界面。

“和欣”的图形构件系统利用 CAR 构件继承机制，设计了一套方便用户自定义开发的构件平台，图 2 就是该自定义图形构件库已经实现的一个框架图。

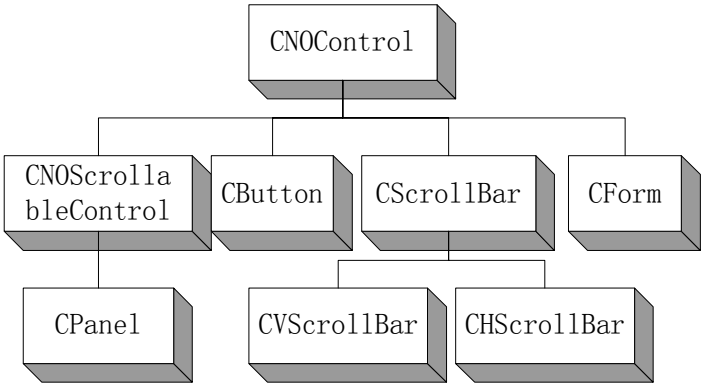

图 2 “和欣”自定义图形构件框架

其中构件类 CNOControl 实现了图形构件的基本控制函数，包括设置控件的大小、位置、颜色、字体等基本信息，以及对控件的常用操作。构件类 CNOControl 提供了 *virtual* 类型的接口 IControl，因此对控件的这些基本操作可以被其他用户自定义构件通过继承接口 IControl 的方式来调用，大大减少了重复

编码和实现，达到构件级的软件复用。

比如要实现一个 CForm 构件，在其构件定义的时候可以这样编写其.cdl 文件：

```
[uuid(0000127a-0000-0000-C000-000000000066), aggregated]
class CForm : CNOControl{
    interface IForm;
    icallback IFormCallback;
protected:
    interface IFormHelper;
};
```

这里新的构件类 CForm 继承了 CNOControl 类，那么其接口 IForm 就可以直接重载和调用 CNOControl 类的接口 IControl 已经实现了的各种控件操作方法。另外，可以看到 CForm 类还声明了一个 protected 类型的接口 IFormHelper，这样如果需要另外实现一个类似 Form 的构件的时候，就可以用 CForm 类派生出新的构件类，并且继承 IFormHelper 接口的方法。由于该接口是 protected 类型的，就只能被派生构件内部使用，不能被外部构件访问。

在这样一个图形构件系统里面，可以充分利用已有的构件来实现用户自定义的构件，通过接口的继承和重载是可以非常方便的达到这样一个目的，从而可以非常方便地实现构件的复用，为图形构件库的丰富提供了一种有效的方式。

4 结 语

CAR 构件编程模型在 COM 技术的基础上，实现了构件二进制继承的创新，其构件继承机制有效的保证了更加灵活和方便的软件复用。因此，CAR 的构件继承技术对于实现软件工厂化生产、提高软件生产效率和软件产品质量具有非常重要的意义。

最后，还需要强调一下，CAR 构件继承实际上是接口级的二进制继承，实现了构件运行时的多态。CAR 的构件继承机制与 COM 的包容、聚合是并列的，都是构件复用的方式。CAR 的构件继承机制不是独立存在的，它与 CAR 的构件封装、自描述特性等技术一道，才能真正成为软件工厂化生产的利器。

参考文献：

[1] 孟德斌,张雪松. 软件重用技术和开发管理模式研究[J]. 计算机工程, 2002, 28（3）: 83-84.

[2] 徐正权,张颢. 基于构件复用的软件方法与 COM 支持[J]. 计算机工程与应用, 2001, 37（9）: 49-50.

[3] 上海构件库. 构件相关简介[EB/OL].
http://www.sstc.org.cn/AboutComponent/CompIntro/CompIntroduce.aspx

[4] 科泰世纪科技有限公司. 《和欣 1.1 资料大全》[EB/OL].
http://www.koretide.com.cn/download.php?DownLoadID=3

[5] 潘爱民. COM 原理与应用[M]. 北京: 清华大学出版社, 1999. 92-94.

[6] 王希辰. 分布对象技术与软件复用[J]. 计算机系统应用, 2001, 6 ：48-50.