# 从 C++多态到 CAR 多态研究软件复用的发展

杨亚俊[1]， 陈 榕[2]， 杨青松[1]
（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）

**摘 要：**随着网络技术的发展以及软件复用技术的日益突出，WEB 服务（Web Service）的概念正在成为新一代因特网应用的重要特点。实现网络服务的关键技术是面向构件、中间件的编程技术，以及一整套的运行环境、开发环境等平台技术。而对 COM 技术进行了扩展的 CAR 构件、中间件技术，则更好地适应了面向 WEB 服务的要求。文章首先分析了传统面向对象语言 C++的仅仅源代码级的多态复用技术，然后发展到二进制代码级标准的 COM 组件技术，通过接口来实现对象的多态性。最后，介绍了由北京科泰世纪有限公司自主研发的与 COM 兼容的 CAR 构件技术，它是面向构件编程的编程模型，实现了构件创建的多态性等先进技术。

**关键词：**软件复用;C++;COM;CAR;多态;组件;接口
**中图分类号：**TP311.52

# A Study On The Development Of Software Reuse from C++ To CAR Different Used In Polymorphism

YANG Ya-jun[1]，CHEN Rong[2]，YANG Qing-song[1]
(1.Software Engineering Center of Graduate School at Shenzhen, Tsinghua University, Shenzhen , 518055, China;
2.Research Institute of Information Technology Operating System and Middleware Technology R&D Center,
Tsinghua University, Beijing, 100084, China)

**Abstract:** With the development of computer network technology and software reuse technology , web service is becoming one of the most important characteristic. The kernel technology of Web Service includes component and mid-component oriented programming, a set of running environment and developing environment technology. CAR technology compatible with COM is more suitable for the request of web service. This paper bases on a thorough analysis of several techniques. First, this paper analyzes the form and characteristic of C++ language's polymorphism. Second, it analyzes MS COM which implements polymorphism through interface. And then, it introduces CAR component technique which was developed completely by Koretide corp. CAR is a kind of component oriented programming model and it implements polymorphism of component creating.

**Key words:** software reuse; C++; COM; CAR; polymorphism; component; interface

## 1 引 言[1]

随着计算机网络技术和构件产业化的发展趋势，软件复用技术越来越被重视，并且有了长足的发展。所谓多态性，简单的说就是“同一接口，多种实现”。多态性(polymorphism)一词来源于拉丁语 poly(表示多的意思) 和 morphos(意为形态) ，其字面的含义是多种形态。它反映了人们在求解问题时，对相似性问题的一种求解方法，面向对象方法的一个重要特点就是它支持软件开发过程中更大的灵活性和更多的可重用性, 而这种灵活性和可重用性主要来源于实体的多态性, 即允许一个实体在不同的环境中以不同的形态出现。多态思想允许同一实体在不同的环境中具有多样的行为解释，因而发挥出不同的功能，具有灵活丰富的描述表达能力，支持数据抽象，有利于自然方便的进行软件设计。

在 C++中，多态性只体现在源码级的多态。多态性是指可以用同一名字定义功能相近的不同函数。因此，多态性又叫做“同一接口，多种方法”。C++用函数重载和运算符重载来实现编译时的多态性。用派生类和

**基金项目：**国家高技术研究发展计划（863 计划）项目（编号 2001AA113400）。 **收稿日期：**2004-07-27

**作者简介：**杨亚俊（1979-），男，山西运城人，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中心副主任，科泰世纪科技有限公司首席科学家，主要研究方向：网络操作系统，构件技术；杨青松，男，硕士研究生，主要研究方向为计算机应用技术。

虚函数来支持运行时的多态性。按 Gardelli 和 Wegner 的分类方法， 多态性可分为以下 4 种（如图 1 所示）：

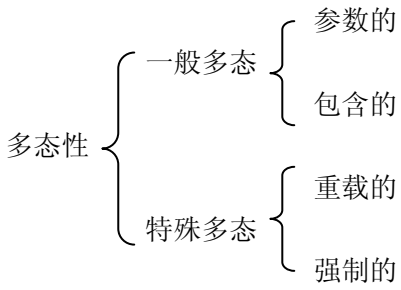


**图1 多态性分类**

传统的面向对象编程语言中，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，就变得毫无意义了。更为重要的是，连编以后，类构件就只是一个逻辑上的虚幻的概念了，不会给将来可能的处理带来任何方便。微软提出的 COM 标准，正是关于如何建立组件及如何通过组件建立应用程序的一个规范，通过 COM 可以建立起一个软件模块同另一个软件模块之间的连接，2 个模块之间就通过称为“接口”的机制来进行通讯。

COM对象对对象上的数据实现完全封装，外部如果想访问对象的数据，必须通过COM对象提供的接口方法。接口完全封装了内部功能，属性的具体实现，使得COM对象对外表现为“黑盒”结构。而C++对象的数据有可能被外部直接访问。C++语言对类的成员进行了访问控制，public成员可以在对象外部直接访问。

COM对象的可重用性表现在COM对象的包容和聚合，一个对象可以完全使用另一个对象的所有功能；而C++对象的可重用表现在C++类的继承，派生类可以访问其父类的非私有属性、调用非私有成员函数。虽然C++和COM的重用性机制不同，但我们可以把这两种技术有机结合起来：在源代码级使用C++的类重用性，在组件一级使用COM的重用性。

C++对象通过虚函数实现多态性。而 COM 对象的多态性通过接口来体现，一个对象可以对应多个接口，一个接口也可以由多个对象所实现，表现出灵活的多态性。同时也为版本维护提供了方便，当使用新版本的组件替换旧版本时，只要该组件实现了旧版本的接口（通过包容，聚合等方式），就保证了它跟原有软件系统的兼容。新的接口功能也可以被自然方便的使用。

通过对COM和C++在多态性方面的简要回顾和比较，我们知道，COM是一种技术，但也可以说是一种编程思想，COM技术的出现使应用程序的开发、维护、扩展更加容易。在COM组件框架的应用程序中，应用程序由组件拼装而成，由于COM组件的语言无关性、接口的多态性和不变性，使得任何语言开发的COM组件都可以被用不同语言开发的应用程序调用。随着COM技术的不断完善，COM技术将得到更广泛的应用。

CAR 技术则是在很大程度上借鉴了 COM 技术，保持了和 COM 的兼容性，同时对 COM 进行了重要的扩展，实现了构件创建的多态。CAR 技术是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，为更好地支持面向 Web Service（WEB 服务）的下一代网络应用软件开发而开发的。

## 2 CAR 构件技术

### 2.1 CAR 技术的由来

80 年代以来，目标指向型软件编程技术有了很大的发展，为大规模的软件协同开发以及软件标准化、软件共享、软件运行安全机制等提供了理论基础。其发展可以大致分为以下三个阶段：面向对象编程、面向构件编程、面向中间件编程。

由于因特网的普及，构件可来自于网络，系统要解决自动下载，安全等问题。图 2 所示即为中间件运行环境的模型。由此可见，系统中需要根据构件的自描述信息自动生成构件的运行环境，生成代理构件即中间件，通过系统自动生成的中间件对构件的运行状态进行干预或控制，或自动提供针对不同网络协议、输入输出设备的服务（即运行环境）。中间件编程更加强调构件的自描述和构件运行环境的透明性，是网络时代编程的重要技术。其代表是 CAR、JAVA 和.NET（C#语言）。

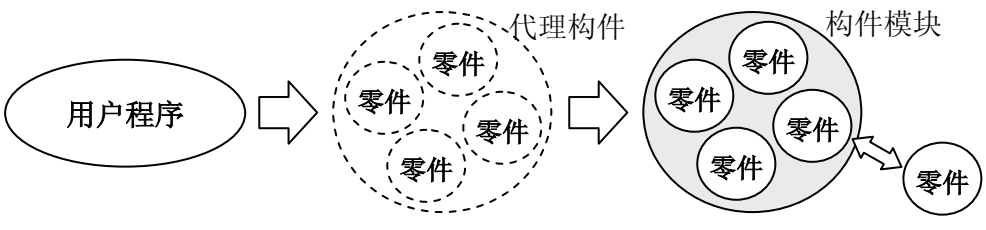

图 2 中间件运行环境的模型，动态生成代理构件

CAR 构件技术的的全称是 Carefree Application Runtime，它是北京科泰世纪科技有限公司开发的面向构件编程的编程模型，也是科泰公司整个“和欣”技术的精髓，它贯穿于整个技术体系的实现中。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 兼容微软的 COM。但是和微软 COM 相比，CAR 删除了 COM 中过时的约定，禁止用户定义 COM 的非自描述接口；完备了构件及其接口的自描述功能，实现了对 COM 的扩展；对 COM 的用户界面进行了简化包装，易学易用。

从上面的定义中，我们可以说 CAR 是微软 COM 的一个子集。CAR 很大程度地借鉴了 COM 技术，保持了和 COM 的兼容性，同时对 COM 进行了重要的扩展。

为了在资源有限的嵌入式系统中实现面向中间件编程技术，同时又能得到 C/C++的运行效率，CAR 没有使用 JAVA 和.NET 的基于中间代码—虚拟机的机制，而是采用了用 C++编程，用和欣 SDK 提供的工具直接生成运行于和欣构件运行平台的二进制代码的机制。用 C++编程实现构件技术，使得更多的程序员能够充分运用自己熟悉的编程语言知识和开发经验，很容易掌握面向构件、中间件编程的技术。在不同操作系统上实现的和欣构件运行平台，使得 CAR 构件的二进制代码可以实现跨操作系统平台兼容。

### 2.2 CAR 技术的特点

CAR 是在 COM 技术的基础上进行改进和扩展的，相对于传统的 COM 技术，CAR 具有以下特点：

1) 易学易用。CAR 对 COM 进行了概念上的简化。CAR 对 COM 的 IUnknown、IDispatch、IConnectPoint 等等 COM 基础接口及源接口、可连接对象等概念在系统级进行了封装，使构件的编写者及使用者不需要过多关心除构件功能以外的其他实现细节，并直接提供对可描述构件、可聚合构件、事件机制等等高级特性的支持，大大降低了 COM 门槛。
2) 对 COM 技术进行扩展。CAR 在秉承了 COM 的核心精神的基础上，在构件的自描述封装及运行、构件类别、构件的自描述数据类型支持等等方面进行了扩展。由此实现了构件的无注册运行、支持构件对象创建的多态性、任意接口的远程化支持等特性。
3) 摒弃了 COM 中一些错误或无关紧要的概念和实现。如 COM 的套间结构及线程模型设计存在自相矛盾的地方，在 CAR 中就摒弃了套间的概念和 CoIinitilize 等相关调用，把对线程模型的支持能力交由构件的开发者决定，使得构件的设计更符合 COM 本身的自描述理论及面向对象的封装思想。

### 2.3 CAR 技术在“和欣”技术体系中的作用

CAR 构件技术是科泰公司“和欣”技术体系的精髓，它贯穿于整个技术体系的实现中。可以说“和欣”操作系统是基于 CAR 实现的，同时它也是全面面向 CAR 构件技术的。

- 构件化的操作系统内核，以及“灵活内核”体系结构；
- 构件化的图形系统、设备驱动、文件系统等操作系统的系统服务；
- 支持应用软件跨平台二进制兼容的“和欣构件运行平台”；
- 构件化的应用软件，支持 WEB 服务，支持动态加载。

在面向嵌入式系统应用的操作系统中全面导入构件技术，对于开发功能越来越复杂的应用系统有着重要的意义。功能丰富的嵌入式设备将更多地要和网络发生关系，要支持 WEB 服务，对于嵌入式系统开发来说同样是重要的。

## 3 CAR 技术对 COM 的发展

CAR 在秉承了 COM 的核心精神的基础上，在构件的自描述封装及运行、构件类别、构件的自描述数据类型支持等等方面进行了扩展。由此实现了构件的无注册运行、支持构件对象创建的多态性、任意接口的远程化支持等特性。

对微软的 COM 进行了扩展主要体现在以下几个方面：

**（1）自描述数据结构**

为支持构件化编程而设计的自描述数据结构：EzStr、EzByteBuf、EzStrBuf、EzArray。EzStr 一般用来存储用户的常量字符串，它有一个定长的存储区，可以存储用户的字符串，它还保存该字符串的长度。EzByteBuf 提供存储字节的缓冲区，可以存放任何数据，EzStrBuf 中存放的是一个 EzStr 对象，EzArray 用来定义一个多维、定长、自描述数据类型的数组。

**（2）与创建、管理构件相关的定义与函数的实现**

主要有：IID_INTERFACE_INFO、ClassInfo、CoInitialize、CoInitializeEx。

**（3）构件类别（Category）与多态性**

### 3.1 以 EzArray 自描述数据结构为例

EzArray 用来定义一个多维、定长、自描述数据类型的数组，它的内存结构如图 3 所示：

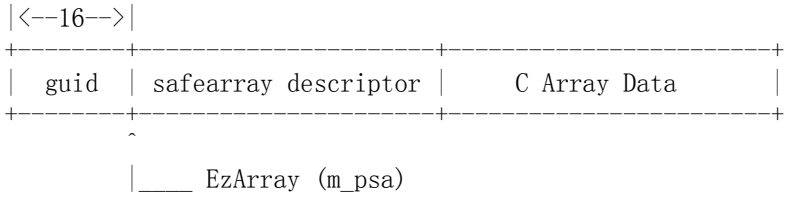


图 3 EzArray 自描述数据结构

EzArray 是对 MS COM 的 SAFEARRAY 的扩展。EzArray 在 ezCOM 中，只有 C++实现，它被定义为一个类。该类有一个成员变量 m_psa，该变量被定义为 SAFEARRAY 的指针类型，有关 SAFEARRAY 结构体的描述参见 MSDN。

我们在 SAFEARRAY 前面增加了 16 个字节，用于存放 GUID。

EzArray 变量既可以分配在栈上，也可以分配在堆上。通过 EzArray_<T, SIZE> inArray 则可以在栈上定义一个数据类型为 T，大小为 SIZE 的数组 inArray。 例如：EzArray_<int,5> inArray 在栈上定义了一个名为 inArray 的整型数组，数组长度为 5。

### 3.2 以 ClassInfo(Class Information)定义为例

在 MS COM 中，用户编写 ODL，然后用 MS 的工具 MIDL 或者 MkTypLib 产生 Type Library 信息，该信息是描述构件的元信息。

在 ezCOM 中，开发了自己的 CDL 语言，关于 CDL 语言，参照“CDL 语言”部分，并且开发了自己的描述构件的元信息，称之为“ClassInfo”。同时开发了自己的编译器工具，使用该工具可以根据 CDL 文件得到 ClassInfo。

### 3.3 构件类别（Category）的原理简述

在 COM 中，所有的构件都是以类标识（CLSID）作为构件类的唯一标识（以下简称 CLSID），每一个 CLSID 就对应了一个构件实现。

COM 规范声称，把虚接口抽象出来，就是实现了二进制的多态。但实际情况并不完全如此：构件客户端在使用构件时，还是要指定构件服务器的 CLSID 来创建构件对象，指定 CLSID 实际上就是指定了构件的实现。

所以我们认为，把接口抽象出来只是实现了构件方法调用的多态性，并未实现构件创建的多态性。而一个构件的使用总是要经过创建、调用、消亡这三个过程的，只有实现了构件对象创建的多态性，才算得上是完整的实现了构件使用的多态。

为了达到上述目的，我们引入了构件类别（category）的概念，一个构件类别包含了一组公用接口，某个构件类（class）要属于这个类别就必须继承这个类别并实现这个类别包含的所有接口。构件类别本身没有实现代码。

因此，构件类别也可以被称为抽象类或超类。

继承于构件类别（category）的所有构件类（class）都要实现构件类别中包含的所有接口，这也正是构件类别作为类别中所有构件类的公共入口点的技术基础。

### 3.4 对构件类别的多态方式支持

使用构件类别实现多态方式创建构件对象的步骤如下：

***步骤一：定义构件类别***

我们可以在 CAR 的 CDL 文件中使用构件描述语言定义类别，如：定义一个打印驱动类别，名叫 CatPrinter，定义方式如下：

```
[ uuid(02aff0b1-a887-4da7-bf2e-626af6165a56) ]
category CatPrinter {
interface IPrinter;
}
```

例子中的 CatPrinter 包含了一个接口：IPrinter（关于接口的概念及定义方式请参见相关 COM 规范）。实际上，一个类别中允许定义一个到多个接口。

***步骤二：定义构件类***

如果某个构件类要属于 CatPrinter 这个类别，那么它必须继承 CatPrinter，如：HP LaserJet 6L 打印驱动构件在 CDL 中的定义方式可以如下例所示：

```
[
```

*uuid(9defd903-6443-4eed-b4e4-a3020b448cb5)，*

*driver*

*]*

*class CHPLaserJet6L : CatPrinter {*

*interface ILaserJet6L;*

*}*

类 CHPLasterJet6L 的定义本身包含了 ILaserJet6L 接口，同时由于它继承了 CatPrinter 这个类别，它还包含了 CatPrinter 中的 IPrinter 接口。

此外，在 class 前的 driver 属性声明说明了 CHPLaserJet6L 是一个驱动程序。它还必须实现别外一个隐含的系统接口：IDriver，因为驱动程序本身也是一个类别，系统中所有以 driver 属性定义的构件类都属于驱动程序类别：CatDriver，IDriver 接口正是在 CatDriver 中包含的。

***步骤三：实现构件类***

使用 CAR 的 CDL 编译器编译定义驱动构件类别及构件类的 CDL 文件，会产生构件的源程序框架。框架中包含了构件类所有接口的方法，构件实现者根据构件的功能需要填写这些方法的实现代码。

以例子中 Cprinter 为例，就必须实现 IPrinter、ILaserJet6L 及 IDriver 三个接口中的所有方法。

***步骤四：编译构件程序并自动注册构件类别***

代码实现完成后，需要使用 CAR 的 zmake 对构件程序进行编译生成构件 DLL 文件。

使用 zmake 编译 CDL 文件及构件源程序时，zmake 会自动把 CDL 中定义的构件类别及类别所包含的构件类注册到系统中。

**注：重复步骤二到步骤四，就可以定义并实现多个属于 CatPrinter 的打印驱动构件类。**

***步骤五：编写构件客户（应用程序）程序***

构件程序编译生成后，就可以编写构件客户程序来使用构件提供的功能了。构件客户程序可以通过两种方式来创建驱动对象，一是采用常规的 COM 创建对象的方式：通过指定具体驱动构件的 CLSID，如：CLSID_CHPLaserJet6L；二是使用类别标识（CATID），如 CATID_CatPrinter。

要想使构件对象的创建具有多态性，必须使用第二种方式，在 CAR 中，可以使用类别的智能指针来创建驱动对象，如：通过类别 CatPrinter 创建打印驱动对象的代码如下：

*#import <printer。dll>*

*……*

*CatPrinterRef catPrinterRef;*

*hr = catPrinterRef。Instantiate();*

*if (FAILED(hr)) {*

*……*

*}*

*……*

***步骤六：运行构件客户程序创建驱动对象***

使用 zmake 编译构件客户程序并在“和欣”操作系统上运行，以创建并使用指定类别的对象。通过类别创建对象的实现过程如下：

1) 指定构件类别标识（CATID）来创建驱动对象；
2) 取该类别的缺省（default）CLSID；
3) 使用 CLSID 创建具体的构件对象；
4) 从对象 QI 出类别中的公用接口；
5) 返回类别公共接口；
6) 构件客户在得到这些公共接口后，就可以调用这些公共接口的方法了。

**3 结论与展望**

从传统 C++源码级的软件复用到 COM 二级制标准的软件复用，组件技术增强了应用软件的灵活性，降低了软件开发与维护的成本。科泰世纪有限公司研发的 CAR 构件技术是面向构件编程的编程模型，它兼容微软的 COM 并对其进行了扩展，具有更多方面的优势，支持构件对象创建的多态性只是其中很重要的一个方面。

因特网的发展经历了以电子邮件、远程文件传输等简单应用为代表的第一代；以网页信息浏览、检索为代表的第二代，第三代因特网的核心是使用“WEB 服务”（Web Service）。主流软件程序设计模型的发展经历了以结构化编程为代表的第一代；以面向对象编程为代表的第二代，第三代编程模型的核心是设计“WEB 服务”。通用操作系统支持的典型用户体验经历了以字符界面应用为代表的第一代；以图形交互应用为代表的第二代，第三代操作系统的核心是支持“WEB 服务”。也就是说“WEB 服务”不论从应用模型、编程模型，还是从运行模型讲，都是未来 IT 技术发展的焦点。CAR 技术则是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，为更好地支持面向 WEB 服务的下一代网络应用软件开发而开发的，对未来程序开发技术具有重要意义。

**参 考 文 献:**


[1] 蓝雯飞，陆际光,覃俊. C++面向对象程序设计中的多态性研究[J]. 计算机工程与应用 2000，23(8):97-99

[2] 陈榕. 下一代 IT 技术焦点：信息交换[N]. 计算机世界 B12，2002.4

[3] 蓝雯飞. 面向对象程序设计语言 C++中的多态性[J]. 微型机与应用. 2000，25(6):10-12

[4] [美]S. B. Lippman 著.侯捷译. 深度探索 C++对象模型[M]. 华中科技大学出版社，2001

[5] 陈文科，唐志敏. 完善 C++的虚机制以增强其多态性[J]. 计算机研究与发展. 1998，35(7):599-604

[6] Chen Rong. The application of middleware technology in embedded OS[C].In: 6th Workshop on Embedded System, In conjunction with the ICYCS, Hangzhou, P R China, 2001-10

[7] 王斌君，葛玮，王靖亚等. C++语言与软件的多态性——C++与软件工程系列之二[J]. 计算机工程与应用. 1998，34(10):46-49

[8] [美]Dale Rogerson 著.杨秀章，江英译.COM 技术内幕[M].北京：清华大学出版社，1999