

# “和欣”脚本编程环境 SCI 设计与实现

摘要：面向构件编程是在面向对象编程基础上发展起来的，它是提高软件生产效率的重要手段。本文首先介绍“和欣”操作系统和 CAR 构件编程模型，然后就笔者为“和欣”设计的脚本语言通用适配层 SCI，探讨了一种脚本语言与 CAR 构件自动适配的技术，以实现基于 CAR 构件的动态编程。

## 1 “和欣”——灵活内核操作系统

“和欣”操作系统是 863 计划的“基于中间件技术的因特网嵌入式操作系统及跨操作系统中间件运行平台”的重要成果，是一个基于构件的微内核现代操作系统。“和欣”操作系统与一些传统宏内核或微内核操作系统（如 Linux、Windows）的最大区别就是微内核模型与基于构件技术的充分结合，形成了“和欣”操作系统的灵活内核架构模型。灵活内核技术在设备多样化的嵌入式开发环境中所具有的优势十分明显。

“和欣”操作系统是一个微内核操作系统。在微内核操作系统中，系统的一些服务，如文件系统，TCP/IP 协议栈等是作为一个与用户进程同等的进程为其他用户进程提供服务的。因此，在微内核操作系统中，一个主要的问题就是进程间的通讯问题。传统的一些微内核操作系统（如 Mach 和 Minix）采用消息机制作为系统服务进程和用户进程通讯的手段。“和欣”操作系统则是采用了更加现代的面向构件编程技术。在面向构件编程技术中，不同进程甚至是不同主机之间的构件通讯是一个基本问题。在应用领域中，也已经有了很多成熟的解决方案，如微软公司的 DCOM（分布式构件对象模型）和 Java 中经常使用的 RMI（远程方法调用）机制。“和欣”操作系统将构件的跨进程（在“和欣”操作系统中称为 Domain，不同的 Domain 不光是指不同的进程，还包括不同的主机）通讯，作为一个操作系统的基础设施，做到了内核之中。从语义上来说，构件的进程间通讯与消息机制等价，但是，基于构件的通讯技术更加符合现代程序设计技术和规范，使得程序更容易模块化，更健壮，而且构件更容易扩充和更换，实现灵活内核。

批注 [x1]: 是这样吗？

“和欣”操作系统中使用 CAR 构件系统，CAR 构件具有充分的自描述元数据，具体细节在下一节介绍。这里只需要说明根据元信息能够自动生成构件对象的代理对象和存根对象，构件跨进程的通讯模式如图 1 所示。

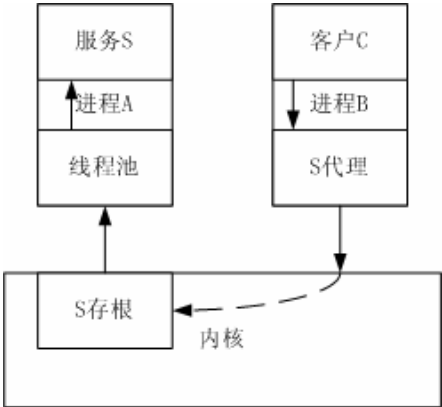


图 1 跨进程的构件服务调用

其中服务构件 S 运行在进程 A 中，首先调用系统调用 EzRegisterService 将服务构件注

册在系统中，在系统中会生成一个 S 的存根对象。客户进程 B 中要使用进程 A 中的 S 服务，客户进程调用系统调用 EzFindService 找到 S 的存根对象，并且在 B 中根据 S 的元数据生成与 S 具有同样接口的代理对象。当客户进程调用 S 的代理对象的方法时，代理对象将方法的调用通过系统调用 SysInvoke 传递给 S，S 所在的进程 A 会启动一个线程调用被请求的 S 的方法。为了优化这种情况，“和欣”操作系统内部给每一进程都附加了一个线程池。

## 2 CAR 构件系统

上文已经介绍了“和欣”操作系统对构件编程的支持，而这种支持又必须以与之配套的构件系统作为基础。在“和欣”操作系统中就是 CAR（Component Assembly Runtime——构件程序集运行时）构件编程系统。CAR 构件系统有两个主要的特征，一是使用 C++ 语言作为实现语言，这是性能的保证，也是 CAR 构件系统能够作为操作系统基础和在嵌入式设备中使用的基础。二是 CAR 构件具有充分的自描述元数据，这是系统能够自动为 CAR 构件生成存根和代理对象，以及 CAR 构件编程系统可以作为一种中间件系统的基础。这两者结合起来，CAR 构件编程系统就成为了一种高性能的中间件构件系统。

编写 CAR 构件，首先要编写一个构件的接口描述文件——CAR 文件。CAR 文件是 CAR 构件的元数据的文本形式。它描述了 CAR 构件中的接口信息，以及接口中每个方法的信息，包括方法参数、返回值等，同时还描述了接口继承体系的信息。如表 1 所示是一个简单的 CAR 文件。

```
module{
    interface IFoo {
        DoSomething([in] int i);
    }
    class CFoo {
        interface IFoo
    }
}
```

表 1 一个简单的 CAR 文件

CAR 文件经过编译，生成 C++ 代码框架、文档框架和二进制的 CAR 构件元数据。接下来，使用 C++ 语言实现 CAR 构件的具体功能。最后，经过编译和打包生成可发布的 CAR 构件程序集，其中包含了 CAR 构件的代码、数据及自描述元数据。这个过程如图 2 所示。

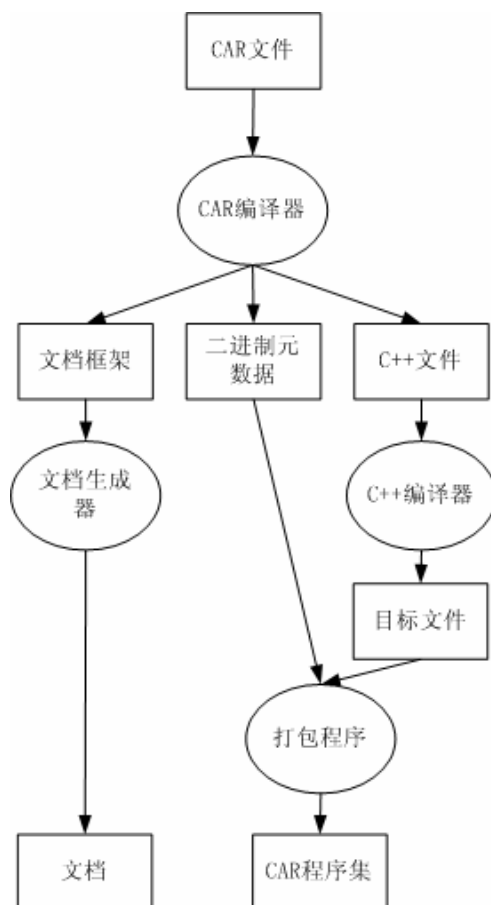


图 2 由 CAR 文件生成 CAR 程序集

本文下面要讨论的脚本调用接口模型的实现也是基于 CAR 构件的自描述元数据。

批注 [x2]: 这个图中需要加入应用编程的环节，否则 CAR 成了整个软件的编程语言了

### 3 脚本调用接口模型

在面向构件的开发模型中，构件完成程序的主要功能和逻辑。大量耗时、性能重要或者复杂的运算被封装在构件内部。构件只需要暴露给外界简单的接口以提供服务。在理论上，面向构件编程中，构件应该是易于使用和更换的，这就极大地提高了应用开发的效率和可扩展性。可是，在已有的一些构件系统中，比如 DotNet 和 Java 中，却往往达不到这种效果。这主要是因为，开发构件和使用构件使用的是同一种语言。开发构件的语言往往是一些通用和功能强大的语言，这就从本质上决定了这些语言的复杂性。这样，使用同一种语言使用构件，往往受语言复杂性的影响，增大了使用构件的难度。并且，这样很容易混淆构件与外部应用的边界，弱化了面向构件的编程思想。

脚本语言通常具有易学易用、开发高效的特点，并且因为脚本语言通常都是解释执行，所以容易更改，不需要重新编译。而脚本语言一般功能较弱，不易扩充。脚本语言的用户通常只能使用脚本语言内置的一些功能。

有些脚本语言（如 Perl、Ruby 等）具有开发扩展库的功能。但通常每种脚本语言有自

己的扩展库开发方法，而且这些方法既不直观也不通用。其中最主要的原因是，出于性能的考虑，这些扩展库都是使用 C\C++语言来编写，可是 C\C++语言并没有一个统一的动态构件能力。

“和欣”设计的脚本语言通用适配层 SCI 的成功开发一定程度上解决了这个问题。利用 CAR 构件中的元数据，动态调用 CAR 构件的方法。CAR 构件的编写人员不再需要考虑如何与不同的脚本语言适配。所有的工作都由脚本引擎 SCI 自动完成。

#### 4 SCI 原理与实现

将脚本语言调用 CAR 构件以获得服务这一要求进行抽象，设计出一个抽象层，从而让脚本语言与构件协同工作。

主要的接口称为脚本可调用接口（Script Callable Interface），下文将简称为SCI。SCI本身也是用CAR构件编程技术实现的。结构图如图 3 所以。

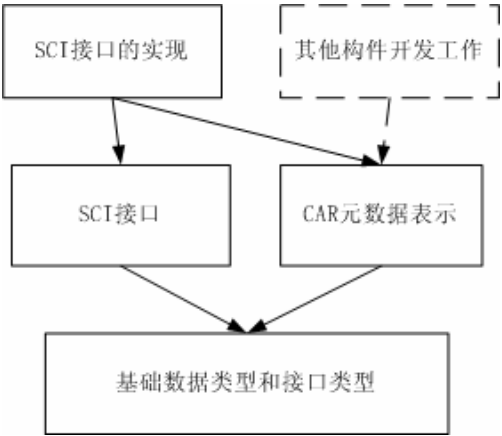


图 3 SCI 系统结构

从图中可以看到，以 CAR 构件的基础数据类型和接口类型为基础，而这些内容恰好是 CAR 构件的元数据所能表示的内容。SCI 的接口以此为基础，实现出对构件方法调用的抽象。SCI 的实现则以对应的元数据为基础。这种结构较好地做到了设计与实现的分离。

SCI接口系统中有 3 个主要接口，它们的关系如图 4 所示。

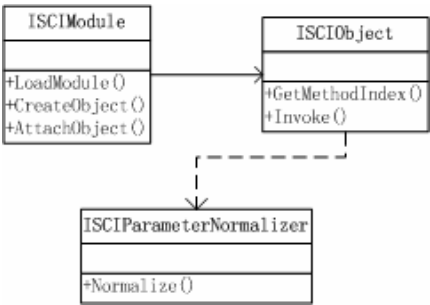


图 4 SCI 接口系统

ISCIModule 负责加载 CAR 构件，从 CAR 构件中集中创建 CAR 对象，（这在“和欣”中称为实例化），并使之与 ISCIObject 对象绑定。ISCIObject 的主要方法为 Invoke，这个方法负责动态调用内部 CAR 对象的方法。

一个方法调用，除了正确的方法入口外，实参与形参的匹配也是一个主要问题。实际工作中的脚本语言千差万别，它们所支持的参数类型和 CAR 构件系统所支持的参数类型通常也并不相同。因此，本文提出了一个参数正规化的概念。参数正规化的过程主要是脚本语言的参数向 CAR 构件参数转化的工程。在脚本引擎通过 ISCIObject 的 Invoke 方法调用 CAR 构件时，脚本引擎将实现 ISCIPParameterNormalizer 接口，由它完成参数转换工作。

## 5 在 Javascript 脚本语言中集成 CAR 构件

Javascript 脚本语言是 Netspace 公司发明的，专门针对 HTML 动态编程的脚本语言。Javascript 是一种面向对象的脚本语言。它的一个主要特点就是对象系统容易扩充。Javascript 脚本语言的实现，选择了 Mozilla 浏览器项目中的 SpiderMonkey 子项目。该 Javascript 的实现具有良好的文档和程序接口。

在 Javascript 脚本语言中集成 CAR 构件，首先，我们在 Javascript 中添加全局对象 Elastos。该对象有如下一些方法：

## 6 结语

“和欣”操作系统是 TD-SCDMA 这一中国人第一次拥有的重大通信国际标准的配套工程，将被用在 3G 手机中，通过本文的工作，将使更多的人可以通过构件的组成，完成一个符合自己要的“应用程序”，从而提高 3G 网络的应用水平。

本文所述工作还有很多后续工作，充分利用和欣操作系统的能力等优化工作也在展开。

本文受国家 863 计划“软件重大专项”支持（课题名称：基于中间件技术的因特网嵌入式操作系统及跨操作系统中间件运行平台，课题编号：2001AA113400，所属专题：计算机软件，所属领域：信息技术领域）。