

# 和欣嵌入式操作系统的XML-Glue编程模型

陈果, 陈榕

(同济大学 计算机科学与工程系, 上海 200092)

**摘要:** “和欣”嵌入式操作系统是基于CAR构件技术<sup>[1]</sup>、支持构件化应用的操作系统, 是国家 863 支持的TD-SCDMA的操作系统标准。文中分析了当前嵌入式应用开发的模式, 提出了基于CAR构件技术的脚本语言加CAR构件的XML-Glue编程模型的设计, 是一种嵌入式操作系统中更适应开发应用需求的编程模型。一方面充分利用了现有系统的特点, 另一方面使得开发简单高效, 推动了构件化编程思想在嵌入式开发中的应用。

**关键词:** 和欣嵌入式操作系统; 元数据; CAR 构件技术; 脚本语言; XML-Glue 编程模型

## XML-Glue programming model in elastos

Chen Guo, Chen Rong

(Department of Computer Science and Engineering, Tongji University, 200092, China)

**Abstract:** Elastos is an embedded operating system based on CAR component technology, and the standard TD-SCDMA operating system supported by the "863" Program. In this paper, the current development model of embedded applications was analyzed, and the XML-Glue programming model composed of scripting languages and CAR was presented, which was more suitable for the needs of application development in embedded OS. It utilizes all the features of existing systems, makes the development of applications easier and more efficient and promotes the understanding and use of component oriented programming.

**Key words:** Elastos embedded operating system; Metadata; CAR; Scripting language; XML-Glue programming model

## 0 引言

面向构件编程是 20 世纪 90 年代中后期发展起来的一种软件开发模型。和欣嵌入式操作系统所开发的 CAR 构件技术在吸收面向对象编程和面向构件编程技术经验的基础上, 更好的支持面向以 Web Service 为代表的下一代网络应用软件的开发。它从根本上改变了传统操作系统的程序组织形式, 充分利用了构件技术模型的优良特性。

当前, 在诸多的嵌入式操作系统上开发应用的模式跟传统的桌面应用开发方式相差无几, 基本上采用强类型的系统编程语言(如 C、Java)及其相应的开发模式来进行开发。随着嵌入式设备的逐渐普及, 特别是移动设备, 这些设备上的应用的需求量远远大于桌面应用的需求, 应用的变化速度也相当的惊人。传统模式的开发周期长的弱点就暴露了出来, 因此在应用的开发中应该引入新的开发模式来适应新形势的发展。

我们在基于CAR构件技术的基础上, 利用现有的脚本语言的优势, 提出了新的XML<sup>[2]</sup>+CAR+其他脚本语言(JavaScript<sup>[3]</sup>、Ruby<sup>[4]</sup>等)的XML-Glue编程模型, 为将来的嵌入式设备乃至桌面上的应用提供了更加快速的开发模式。

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）  
陈果（1982-），男，四川成都人，硕士研究生，主要研究方向：系统软件支撑技术；陈榕(1952-), 男，教授。

## 1 关于和欣嵌入式操作系统和 CAR 构件技术

### 1. 1 和欣嵌入式操作系统

和欣嵌入式操作系统是一个基于构件化软件模型的系统,构件化软件设计思想贯穿了整个系统的设计与实现中,系统实现本身就是构件模式。除内核中最底层的控制部分外,所有系统功能都是以构件接口的形式提供。另外,操作系统对构件化软件模型提供了必要的运行环境,来源不同的构件可以在该环境上实现互操作。系统提供了构件自动寻址/自动加载机制,用户不必知道调用的构件程序是本地的还是来自于网上,也就是说,构件运行环境可以对用户透明。构件化系统的实现,使得操作系统本身具有高度的灵活性和扩展性。和欣采用的构件技术是 CAR 技术。

### 1. 2 CAR 构件技术

CAR(Component Application Run-Time)是一个国内的自主知识产权的构件系统,是由上海科泰世纪有限公司开发的新一代的构件系统。其主要目的是从操作系统层面上引入构件的概念,所有的服务由构件来提供,实现软件的目标代码级的重用,是新一代的构件系统,为网络编程和 Web Services 提供了强大的支持。CAR 是一个面向构件的编程模型,它表现为一组编程规范,包括构件、类、对象、接口等定义与访问构件对象的规定。

## 2 XML-Glue 编程模型

### 2. 1 XML-Glue 编程模型简介

Glue, 1) 名词性, 胶水; 2) 动词性, 粘合。XML-Glue 形象的指明了它在和欣构件平台中扮演的角色的主要特征, 即粘合剂, 而它所粘合的对象就是一个个构件, 它的粘合可以使功能分立的构件组合成一个可以向用户提供完整功能的应用实体。从另一方面说, 由于它的存在, 消除了构件间的耦合, 使构件只需符合 XML-Glue 的接口标准, 就能和其他构件协同工作。

更进一步讲, XML-Glue 技术是一种架构支持技术。

软件体系结构通常被称为架构 (Architecture)。目前架构理论仍在发展阶段。软件体系结构问题包括软件各个方面的组织和全局控制结构, 通信协议、同步, 数据存储, 给设计元素分配特定功能, 设计元素的组织, 规模和性能, 在各设计方案之间进行选择。Garlan & Shaw 架构模型的基本思想是: 软件体系结构={ 构件(component), 连接件(connector), 约束(constrain)}。其中构件可以是一组代码, 如程序的模块; 也可以是一个独立的程序, 如数据库服务器。连接件可以是过程调用、管道、远程过程调用(RPC)等, 用于表示构件之间的相互作用。约束一般为对象连接时的规则, 或指明构件连接的形式和条件, 例如, 上层构件可要求下层构件的服务, 反之不行; 两对象不得递归地发送消息; 代码复制迁移的一致性约束; 什么条件下此种连接无效等。

XML-Glue, 相当于 G&S 模型中的连接件。目前, XML-Glue 支持的架构是 MVC, 使用 XML-Glue 将使具体应用设计人员产生完全符合 XML-Glue 所支持的软件架构下的应用实

例。使用 MVC 架构，最主要的目的是实现 View 和 Model 的分离，支持两者间的独立变化。XML-Glue 使用下面的技术达到对 MVC 的支持：

1. 使用 XML 描述 View；
2. 使用多种脚本语言（JavaScript、Ruby 等）实现 View 和 Model 间数据传递的解耦；
3. CAR 构件技术实现 Model

关于 MVC 模式和 XML 的知识以及模式和架构的关系请参考其他文献。

## 2. 2 XML-Glue 编程模型的目的和目标

XML-Glue 的目的是设计一种新的编程模型来代替传统应用的开发模型。这种新的模型应该具有开发快速便捷，并能灵活配置、按需应变的能力。在向 Web Service 发展的趋势下，CAR 构件作为 Elastos 中服务的提供主体，客户能够更方便的获取所需的服务。

XML-Glue 的目标是通过 XML 把一个应用需要的构件及构件间的关系进行描述，并通过脚本语言实现对构件、构件内的资源及构件运行所需聚集(aggregate)、上下文环境(context)等内容进行操作，从而实现计算环境要求弱化，提高软件网络分布运行能力的一种软件编程技术，是随着 Elastos 这一国产操作系统的诞生而研发的。

## 2. 3 XML-Glue 编程模型的组成及特点

完整的 XML-Glue 编程模型由几部分组成：XML-Glue 语言编程（即描述所使用的 CAR 构件之间关系的一组 XML 标签）、JavaScript 语言、XML-Glue 运行环境(Runtime)、脚本可调用接口对象(SCIObjct, Script Callable Interface Object)和 CAR。它们之间的关系如图 1 所示。XML-Glue 有如下特点：

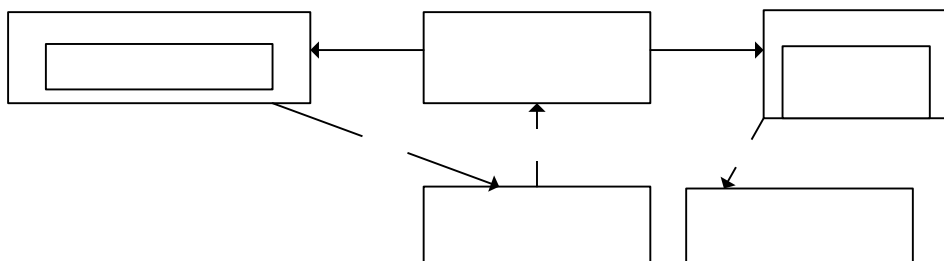


图 1 XML-Glue 架构

- ◆ XML 里嵌入 JavaScript，这就像在 html 里嵌入 JavaScript 脚本一样，是为了弥补 XML 在描述逻辑上的缺陷。XML 标签描述了整个应用的架构，构件及构件间的关系；JavaScript 使得应用能够访问独立于架构的构件及构件内容（接口）的访问，使运行在客户端的 XML-Glue 程序的表现力更强。
- ◆ XML-Glue 的架构通过对多种脚本语言实现(runtime environment)上的抽象，设计与脚本语言的适配器(normalizer, 下面会讲到)，从而允许方便的为它嵌入一种新的脚本语言，如 Python、Ruby 等，适应不同类别程序员的需求和爱好。
- ◆ XML-Glue 解释器并不直接操作所需要的 CAR 构件对象，而是通过一个 SCIObjct 对象来进行计算。一个 SCIObjct 对象封装了具体它要进行计算的 CAR 构件对象，分离了各种脚本语言操作 CAR 构件过程的相似和不同之处，将相似的地方抽取出来（比如 invoke 一个 CAR 构件方法的过程等），使得不同脚本语言的解释器只需要处理各自语言不同的地方（比如脚本的解析等），再调用 SCIObjct 的统一接口即可完成计算。这就使得在 XML-Glue 中对多种脚本语言的支持成为可能。
- ◆ 调用 SCIObjct 接口进行 CAR 构件方法的调用的过程中，需要将脚本语言中的弱

数据类型参数(通常是字符串类型)转变为 CAR 中的强数据类型参数(int、double、EzStr 和接口类型等)。为了适用不同脚本语言的需求, 在实现参数类型转换的时候采用了回调机制, 每一种脚本语言都编写了自己特有的实现了 ISCIParameterNormalizer 接口的类, 在相应的脚本语言解释器中操作 SCIOObject 的时候将相应的类作为参数传入 SCIOObject, SCIOObject 在参数转换时就会根据元数据(meta data) 信息, 计算出需要的参数类型, 然后回调这个类的 Normalize 方法进行类型转换。

- ◆ XML-Glue 中的 MVC 实现模型。前面讲过 XML-Glue 中实现了对 MVC 模型的支持, 组成 MVC 模型的成员关系如图 2:

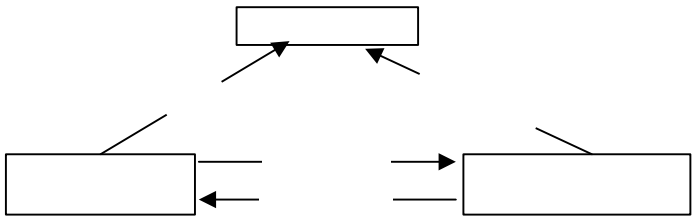


图 2 XML-Glue 中的 MVC 实现模型

从图 2 看出, 除了作为 Model 的 CAR 构件是用强类型的 C++语言编写, 其余的 View 和 Controller 都可以用脚本编写, 简化了开发的难度, 提高了开发效率。

2. 4 XML-Glue 编程模型的应用

我们先来看看 XML-Glue 是如何描述一个应用的。下面是一个简单的代码示例:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <x:xglue xmlns:x="http://www.koretide.com/xml-glue"
3   xmlns:w="elaever.dll">
4   <script language="javascript">
5     <![CDATA[
6       function OnBtnClick() {
7         hello.text = "hello";
8       }
9     ]>
10   </script>
11   <w:form caption="Hello World!" left="10" top="10" width="70" height="50">
12     <w:label x:id="hello" left="0" top="0" width="35" height="20" />
13     <w:button x:id="btn1" caption="Click Me" left="37" top="0" width="70" height="20">
14       <x:event type="Event_Click" handler="javascript:OnBtnClick()"/>
15     </w:button>
16   </w:form>
17 </x:xglue>
```

其运行效果如图 3:

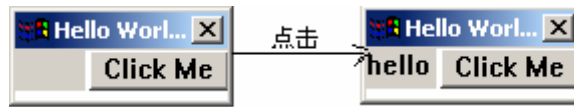


图 3 运行效果图

从这个简单的例子我们可以看出 XML-Glue 编程模型具有以下一些特点：

- ◆ XML-Glue 的源代码文件完全遵循 W3C 的 xml 规范，是规范的 xml 文件。
- ◆ XML-Glue 可以通过名字空间 (xmlns) 在开始部分指定所使用的 CAR 构件 (2-3 行)。在代码的其他部分就可以通过这个名字来访问构件中提供的接口。比如 w:form 就是指 elaevery.dll 构件中的 IForm 接口。
- ◆ XML-Glue 支持直接在源代码中嵌入 JavaScript 等其它脚本语言 (4-10 行)，或者将脚本代码保存成单独文件，在 XML-Glue 中包含进来。这样就能采用其它语言处理逻辑的优势，将一个个独立的构件、对象粘合在一起，使一个个单一的功能拼装出具有强大计算功能的应用，起到了 Controller 的作用。上例中，一个 button 和一个 label 对象就通过一个 JavaScript 函数建立了联系，一起发挥作用。
- ◆ 对于构件属性的设置，可以按照通常 html 中设置属性的方式进行 (11-13 行)，以字符串的形式提供属性值，属性的名称依赖于实际使用的构件的实现。此种访问构件及其属性的方式，是通过 CAR 构件技术对元数据的支持来实现的。通过对元数据的分析计算，能够找到所需的接口、方法的入口点，使得以脚本方式操作构件成为可能。

真实应用中，有多个视图的情况。为了支持多视图 (本例中表现为界面) 应用，XML-Glue 引入 forward 标签，对应了 IXMLGlueApplication 中的 Forward 方法。该方法有两个参数，一个是要跳转页面的 URI，就是上面 XML 文件中的 uri 属性，另一个是 Forward 的类型 (在 xmlglue.car 中定义)：

```
typedef enum ForwardMethod {
    ForwardMethod_Replace,
    ForwardMethod_New,
    ForwardMethod_NewModal
} ForwardMethod;
```

XML-Glue 通过解决下列 3 个问题从而灵活有效的支持了页面跳转：

1. 是否跳转？
2. 跳转到哪里？
3. 以何种方式跳转？

首先，应用程序可以在运行时决定是否进行页面跳转。在 XML 中我们通过 forward 标签所在的 event 标签的 handler 的返回值的真假，来决定 forward 操作是否执行跳转。

第二，应用程序可以在运行时决定是跳转到哪个页面。在 XML 中我们通过 forward 标签的 uri 属性静态指定所跳转的页面。

第三，应用程序可以在运行时决定运行时以何种方式跳转。有 ForwardMethod\_Replace, ForwardMethod\_New, ForwardMethod\_NewModal 三种方式。

可以看出，XML-Glue 提供的编程能力是完备的，完全可以满足应用程序的常见需求，并支持复杂的业务逻辑。

### 3 总结与展望

和欣嵌入式操作系统是面向 SOA<sup>[5]</sup> (Service-Oriented Architecture) 设计的，CAR 构件技

术保证了服务的提供，而XML-Glue编程模型则提供了一种新的使用这种服务的方式。在嵌入式应用千变万化的今天，XML-Glue将应用逻辑和一个个单独的功能分离开，服务提供商只需提供功能独立的正确的构件，应用开发人员就可以根据需求的变化快速更改构件之间的关联关系，降低了服务提供者和使用者的耦合程度。

总的来说，XML-Glue 强化了 SOA 编程的思想，弱化了服务（构件的生产者）和应用（构件的使用者）之间的关系，使得通过它开发的应用能够方便快捷的使用分布在世界各地的服务，不需要所有服务都集中到一起，降低了运行环境的要求，扩大了适应范围。

当前 XML-Glue 与 Elastos2.0 结合的比较紧密，依赖于图形系统和底层的元数据机制。将来在移植 XML-Glue 到其它 OS 上需要解决几个问题：

- ◆ 设计 XML-Glue 图形系统适配层，使得 XML-Glue 操作图形系统有一个统一接口，不同 OS 具体实现对应的接口；
- ◆ 设计XML-Glue元数据操作适配层，COM的元数据单独存放在tlb文件中<sup>[6]</sup>，java的元数据以字节码的形式存放于class文件中，有了适配层，如同图形系统一样，不同的构件机制只需要实现自己相应的接口供元数据适配层操作即可。

## 参考文献

- [1]科泰世纪. 和欣 1.1 资料大全[EB/OL]. <http://www.koretide.com.cn/download/download.php?id=2>
- [2]<http://www.w3.org/XML/Schema>[OL]
- [3]<http://www.mozilla.org/js/>[OL]
- [4] Dave Thomas, Chad Fowler, Andy Hunt. Programming Ruby: The Pragmatic Programmers' Guide, Second Edition. Addison Wesley Longman, 2005
- [5]Munindar P. Singh, Michael N. Huhns. Service-Oriented Computing: Semantics, Processes, Agents[M].Wiley, 2005
- [6]Dale Rogerson. COM 技术内幕-微软组件对象模型[M]. 1999