

# 基于事件机制的嵌入式图形编程模型的研究

郝若梦, 陈 榕

(同济大学 基础软件工程中心, 上海 200092)

**摘 要:** 嵌入式图形系统是嵌入式研究领域的热点问题之一。文中介绍了基于构件技术的 Elastos(和欣)操作系统上图形系统的架构, 该图形系统为应用提供了基于事件机制的图形编程模型。基于事件机制的控件集应用开发有别于基于传统的消息驱动机制的应用程序, 具有简单、高效、对用户透明等特点, 为图形应用开发提供了更多手段, 方便采用 Elastos 的手机开发。

**关键词:** 事件机制; 图形编程模型; CAR

**中图分类号:** TP311.5

**文献标识码:** A

**文章编号:** 1673-629X(2007)10-0001-03

## Research of Event Mechanism Based Embedded Graphical Programming Model

HAO Ruo-meng, CHEN Rong

(System Software Engineering Centre of Tongji University, Shanghai 200092, China)

**Abstract:** Embedded graphical system is one of the hot topics in embedded research field. First, the structure of component based graphical system under Elastos operating system is presented. This graphical system supports an event mechanism based coding mode for graphical applications. The event mechanism based applications on graphical system of Elastos is different from traditional applications based on message drive mechanism. It's simple, effective as well as transparent to customers. It supports more means for graphical applications. It is convenient to develop applications of cellphones on Elastos.

**Key words:** event mechanism; graphical programming model; CAR

## 0 引 言

图形系统是现代操作系统不可缺少的部分, 它为用户的操作提供了极大的方便, 也在某种程度上加快了计算机业的发展。基于传统图形系统的应用(如 Windows, GNOME)大都使用基于消息轮询机制<sup>[1]</sup>的编程模型。随着网络技术的发展, 分布式技术正成为发展的趋势, 又由于现实世界中存在“别调用我, 我来调用您”的模型, 就会产生与传统客户端调用服务端方向相反的“回”调。

文中介绍了国产 Elastos(和欣)嵌入式操作系统上的事件机制, 以及基于该机制的图形系统提供的基于回调接口的事件机制对于其所支持的应用带来的影响。

## 1 CAR 构件技术与和欣嵌入式操作系统

### 1.1 CAR 构件技术

CAR 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, 实现软件的目标代码级的重用。其主要目的是从操作系统层面上引入构件<sup>[2]</sup>的概念, 所有的服务由构件来提供, 从而为网络编程和 Web Services 提供强大的支持。

### 1.2 和欣嵌入式操作系统

和欣嵌入式操作系统<sup>[3]</sup>主要为手机、PDA 等嵌入式终端产品量身定做。它采用 CAR 构件技术, 是一个基于构件化软件模型的系统。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, 系统提供了构件自动寻址、自动加载的机制, 也就是说, 无论是本地或远程的调用对用户都是透明的。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<sup>[4]</sup>。

## 2 事件机制

事件机制是客户端在服务端注册事件, 服务端在

收稿日期: 2007-01-18

基金项目: 国家八六三计划“软件重大专项”(2001AA113400)

作者简介: 郝若梦(1983-), 女, 河北石家庄人, 硕士研究生, 研究方向为系统软件支撑技术; 陈 榕, 中心主任, 教授, 博士生导师, 研究方向为网络操作系统、构件技术。

相应的事件产生后根据注册的内容调用客户端程序的机制。

事件机制实际上体现了一种新的编程模型,结构如图 1 所示。

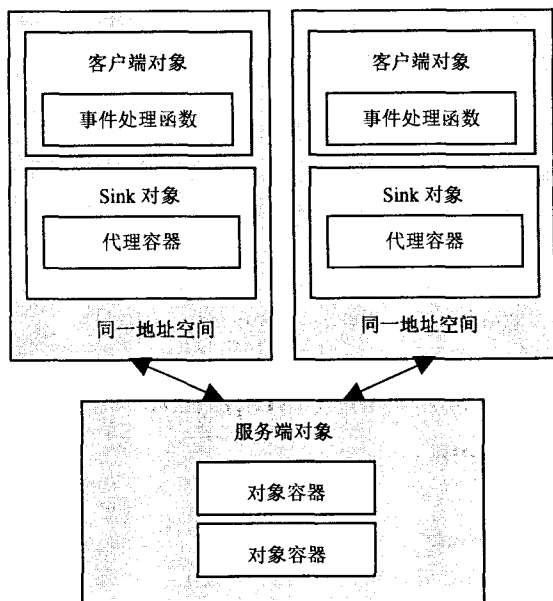


图 1 事件机制的结构

客户端应用程序中定义了事件响应函数(称为 Event 处理函数),并使用相应的方法将 Event 处理函数注册到系统中,即把 Event 处理函数的指针保存在一个 Sink 对象的代理容器中。Sink 对象中有一到多个代理容器,每一个代理容器都是一个代理对象,用链表形式保存回调接口方法的各个 Event 处理函数的指针,每个接口方法有自己的一个队列。

服务端对象容器同样使用链表形式保存 Sink 对象的指针,每个 domain 有一个 Sink 对象,用作服务端的代理。若一个客户端对象需要多个服务端对象的服务,则该客户端对象的 Event 处理函数将被分别保存在多个 Sink 对象中,每个 Sink 对象的回调指针被保存在相应 Server 端的对象容器中。

在调用事件响应函数处理 Event 时,服务端通过链表记录判断 Event 对应的回调接口方法注册在哪个 Sink 对象中,并向其发送消息,该 Sink 对象将遍历相应接口方法的 Event 处理函数队列,执行每一个 Event 处理函数。Elastos 上的事件机制支持 Event 处理函数异步完成,避免某些应用死循环导致的系统崩溃,保证了终端产品的稳定性。

### 3 Vinci 中的编程模型

和欣图形系统 Vinci 是自主研发的图形系统,为智能手机提供 UI 支持,并为第三方的应用开发提供图形支持。

Vinci 图形系统分为物理设备层、系统构件层、应用扩展层三部分<sup>[5]</sup>,如图 2 所示。

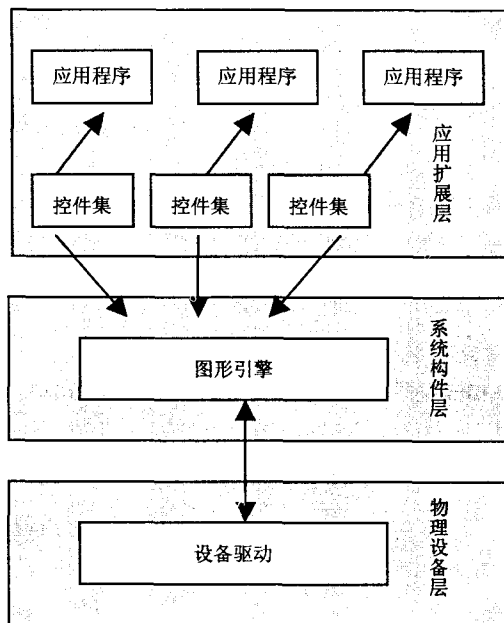


图 2 Vinci 结构图

设备驱动抽象出一些预定义的公共接口,屏蔽了不同硬件的差异。图形引擎就可以利用这些接口函数与硬件交互。

图形引擎提供了图形系统运行所需的最小集合,其中包括字体管理、屏幕管理、图层管理、消息队列管理等。图形引擎向下负责接收驱动传来的消息,并向驱动传送输出图形的信息;向上对用户提供了基本的图形绘制功能,并通过对 Form 的管理进行消息分发。图形引擎通过 CAR 构件的形式向外暴露 API 接口供应用程序调用。

应用扩展层的控件集对底层引擎接口进行封装,实现了一系列的功能。这一层中的一大特色就是用户可以在应用程序中可以自定义事件处理函数,通过和欣操作系统的事件机制,借助控件集的接口与图形系统内核进行交互。

#### 3.1 事件机制在 Vinci 中的实现

和欣操作系统和 CAR 技术为和欣的各个子系统提供了使用事件机制的基础设施,各个子系统则分别实现了自己的事件系统。

在 Vinci 中,控件集的 CAR 文件中的类包含回调接口的声明。回调接口用于向系统注册各个事件的 Event 处理函数。

在 Vinci 中,将控件的共同事件和方法提取出来,形成 IControl 接口,其余的控件接口都继承自 IControl,并包含自身特有的事件和方法。

每个控件的回调接口在图形系统中对应一个对象

容器。应用程序注册的 Event 处理函数被保存在对应控件的 Sink 中。Sink 被保存在对象容器中。当产生一个硬件消息,引擎根据消息参数判断将消息交给某个 Form;作为服务端的父容器 Form 对客户端是否注册了相应的 Event 处理函数并不知情,仅仅通过坐标信息选择相应控件激发一次事件,此时客户端的 Sink 会判断该构件是否注册过 Event 处理函数,如果找到则调用该 Event 处理函数。

### 3.2 事件机制在 Vinci 中的应用

事件机制在图形编程模型中的应用主要体现在图形系统应用扩展层。每一个控件都对应了一组基于控件属性的事件,对于不同的事件,用户可以自定义 Event 处理函数,一方面体现了用户定制功能;另一方面对该控件消息的传送更有针对性。

下面通过一段事件相关的例程来看一下 Vinci 中基于事件机制的图形应用开发步骤。

```
1 CButton::New(&m_pIBtn1);//
2 m_pIBtn1->Init(0, L"Click", 120, 220,
3     80, 30, m_pMainForm);
4 CButton::AddClickEvent(m_pIBtn1,
5     (void *)this, &CButtonDemo::OnClick);
6 .....
6 ECODE CButtonDemo::OnClick(
7     /* [in] * / IButton * pIButton)
8 {
9     pIButton->SetText(L"Clicked");
10    return NOERROR;
11 }
```

由这段程序可以看出基于事件处理机制的图形应用步骤:

(1)创建控件对象并初始化。程序第 1~3 行创建了一个 Button 对象并对它进行初始化。

(2)使用 Add+ 控件事件名 + Callback() 的函数注册该控件的 Event 处理函数。本程序 4,5 行调用 AddClickCallback() 方法对用户程序的 Event 处理函数进行注册,该方法参数中注明了控件的事件类型,Event 处理函数 Handler。本例中,程序为指针为 m\_pIBtn1 的 Button 对应的事件 Event\_Click 定义了函数 OnClick。

在源文件的镜像文件中,CAR 编译器自动生成 AddClickCallback() 的实现代码,如前所述,Event 处理函数的指针被加入 m\_pIBtn1 所指向的回调接口的 Click() 方法队列中。

(3)实现 Event 处理函数。程序 6~11 行实现了函数 OnClick()。

通过该 Button 的 Click 事件进行自定义,可以灵活运用各控件,而且,用户只需将 Event 处理函数通过一个方法调用注册在系统中,并不需要了解事件的过程,整个过程对用户是透明的。

其运行结果如图 3 所示,点击“Click”Button 的显示文字会变为“Clicked”。



图 3 程序效果图

## 4 总结与展望

使用事件机制的图形系统避免了 Windows 应用程序中对大量无用消息的判断,而是根据用户程序注册的 Event 处理函数,有针对性和目的地发送消息,从而节省了大量的时间,更加适用于网络环境。

事件机制中服务端与客户端的划分不仅对用户屏蔽了不同进程不同机器间函数调用的区别,也在一定程度上降低了服务与客户端的耦合度,同时,由于耦合度的降低与接口的固定不变,也方便了模块的升级。

事件机制中由客户端的动作通知服务端调用客户端的 Event 处理函数,为不同机器之间的同步操作提供了方便,因此,在需要信息同步的应用中非常适用,例如一些联机游戏。

文中研究成果已经成功应用于国产 TD-SCDMA 手机中。

### 参考文献:

- [1] Feng Yuan. Windows Graphics Programming Win32 GDI and DirectDraw[M]. 北京:机械工业出版社. 2002.
- [2] Koretide. CAR's Manual[M/CD]. 2004 [2005-06]. <http://www.koretide.com.cn>.
- [3] Koretide. Website[EB/OL]. 2004-04. <http://www.koretide.com.cn>.
- [4] Chen Rong. The Application of Middleware Technology in Embedded OS[C]//Workshop on Embedded System, In Conjunction with the ICYCS(6th). Hangzhou:[s. n.], 2001: 1-3.
- [5] 科泰世纪. Elastos 资料大全[EB/OL]. 2003-09. <http://www.elastos.com.cn>.

邮发代号:52-127, 欢迎订阅。