封闭式手机系统上中间件软件抽象层的设计①

Design of Software Abstract Layer of Middleware on Closed Mobile Phone System

方　军　金　瓯　(中南大学 信息科学与工程学院 湖南 长沙 410083)

摘　要： 本文提出一种架构在封闭式手机系统之上的中间件。中间件屏蔽不同平台之间的差异。应用软件无需再为不同手机平台开发不同版本，从而提高应用软件的开发速度、降低开发成本。本文首先分析了已有的中间件，其次论述了基于封闭式系统的中间件的架构，接着详细阐述了中间件的软件抽象层的设计，最后对中间件给封闭式手机带来的利处进行了说明。

关键词： 功能手机 封闭式手机操作系统 中间件 软件抽象层

1　前言

功能手机和智能手机是面对两个不同的细分市场，在 3G 时代功能手机还是会占领相当大的一块市场份额，而这部分用户同时又是增值业务的活跃用户，功能手机上的中间件在 3G 时代还是会有强大的生命力。

功能手机有着不同的硬件平台，不同的硬件平台又使用了不同的软件平台，比如国产功能手机的主流硬件平台就有 MTK、ADI、展讯、飞利浦、安凯、互芯等，不同平台的软件接口完全不同，程序要实现跨平台移植需要投入大量的人力和物力。

功能手机的 SDK 需要获得厂商授权，小规模公司以及非手机设计公司很难获得厂商授权，这样在功能手机的软件开发上造成了很高的入门门槛。

功能手机操作系统的封闭性造成了程序无法动态更新，除非刷新整个手机 Flash，否则手机上的程序无法更新。当 3G 移动网络来临时，这种封闭性会造成功能手机无法进入利润更大的 3G 增值业务市场。

基于以上几点的考虑，提出运行在功能手机上的一个运行环境，它对不同的手机平台进行了封装，代码只需编写一次即可实现源代码级的跨平台移植。通过在功能手机上预制一个中间件，通过把异构平台的接口进行统一封装，来为应用程序提供一个统一的开发和运行界面。对于使用封闭操作系统的功能手机，中间件能够有效降低手机应用开发的技术门槛，同时中间件也能够提供一条软件分发通道，大量的应用程序能够直接部署到手机中，这对于中小型手机应用开发商来说是至关重要的，这意味着构成游戏和行业应用开发主力的小型企业能够加入到 3G 业务中来。

2　手机中间件分析

目前主要有两种手机中间件，第一种是 SUN 主推的 J2ME(Java 2 Micro Edition)[1]，J2ME 是 JAVA 语言的精简版本，能够加载运行以 JAVA 语言编写的应用程序。J2ME 解决方案是以软件虚拟机来运行 JAVA 程序，所以说它对目标平台的处理器和内存的需求都比较大，虚拟机也需要占用很大的 ROM 空间。

第二种是 Qualcomm 的 BREW(Binary Runtime Environment for Wireless)[2]。BREW 是一个位于嵌入式芯片操作系统之上的抽象层，它能够提供对一些低级功能、环境变量和子程序的访问功能。与 J2ME 不同的是，开发者可以编写 C 代码，该代码将直接为高度受约束环境中的 ARM 处理器进行编译。BREW 只能在 Qualcomm 提供的芯片组(MSM 及 MSP)上运行[3]。

功能手机因硬件配置的限制，在处理能力和内存空间上较智能手机有较大的差距。对于 JAVA 对目标平台的处理器和内存的需求都比较大，使得 JAVA 在功能手机上无法大展身手[4]。虽然 JAVA 对程序员更

① 收稿时间:2008-10-10

© 中国科学院软件研究所 http://www.c-s-a.org.cn

友好，但是虚拟机需要更大的 CPU 和内存来完成计算任务。其沙箱模型也进一步限制了应用程序的可扩展性[5]，因此 J2ME 并不能为手机厂商带来核心竞争力。对于 BREW，因受限于 Qualcomm 提供的芯片组，就无法在使用非 Qualcomm 芯片组的功能手机上应用。

跟 BREW 类似，我们提出一种软解决方案。构建一个支持机器码动态装载的二进制兼容平台 VRE(Virtual Runtime Environment)。相对于第一种中间件 J2ME 来说，这种平台能够直接加载运行编译为机器码的程序，能够以更小的运算量和内存开销来完成任务。相对于第二种中间件 BREW 来说，不受限于 Qualcomm 芯片组。而且直接构建在目标平台上的中间件可以作为一个二次开发平台，在这个基础上能够直接开发出手机专用程序，比如短信管理器、浏览器、防火墙和播放器等内置应用，在终端就表现在消费者欢迎的经常使用的软件功能提升。

3　封闭式手机系统上中间件的设计

VRE 在手机操作系统和应用程序之间起桥梁的作用。VRE 可以包含操作系统提供的所有资源，上层应用不需要关心底层是什么系统，应用的开发直接基于 VRE 平台。

从结构上，VRE 是底层操作系统的扩展，对操作系统做了重新封装，并且 VRE 根据需要可以增加特定功能。VRE 的架构如图 1 所示。

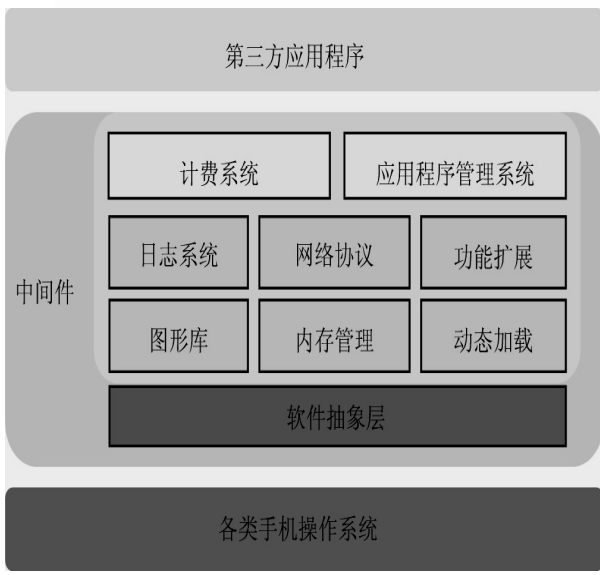

图 1　VRE 架构图

VRE 总的分两层：

(1)软件抽象层(Software Abstract Layer)

要设计好一个中间件，必须将不同手机操作系统功能上的差异，如内存管理、文件系统、图形设备、网络连接等进行整合，封装在软件抽象层中。包括模块有系统事件、输入接口、内存操作、图形设备、TCP/IP 协议栈、时间接口、文件系统等[6]。

(2)核心层

在 SAL 的基础上，扩展功能。核心层的模块是与底层无关的，也可以再分为若干层次，根据功能需要可以包括的主要模块有图形库、内存管理、脚本引擎、动态加载、日志系统、网络协议、窗口系统、计费系统、应用管理系统、其他功能扩展等模块。

VRE 中间件的架构总体采用层次化设计，各层之间仅有单向的依赖关心，层间独立，结构清晰易懂，核心层调用软件抽象层，抽象层调用底层手机系统。

4　VRE软件抽象层的设计

软件抽象层是实现跨平台的核心部分。使用 SAL，各类应用对所有的手机平台来说都是一致的。SAL 提供了可大规模重用代码的设计方式。加入了 SAL 后，它把平台相关部分和平台无关部分隔离开来，这样当该应用支持一种新的手机平台系统时，与平台无关部分就不用进行任何修改，只需修改有关部分就可以了，从而大大提高了代码的重用性。软件抽象层的调用关系如图 2 所示。

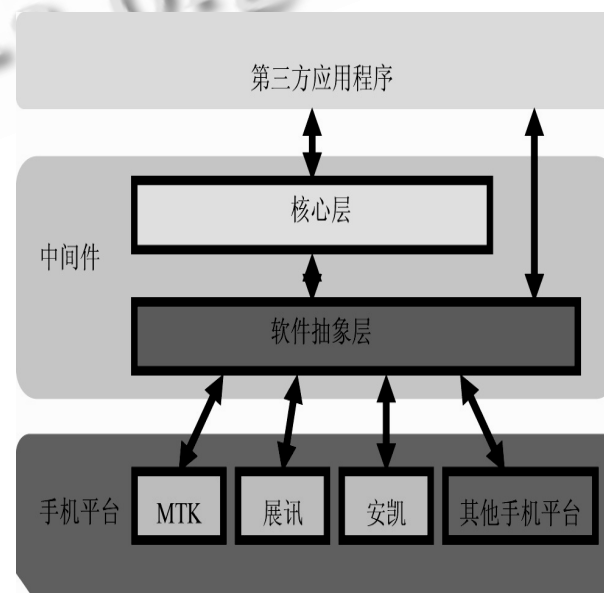

图 2　软件抽象层的调用关系

© 中国科学院软件研究所 http://www.c-s-a.org.cn

抽象层还具有伸缩性。不同的应用可能需要不同的系统调用，如有的需要文件系统，有的不需要文件系统，可根据具体的应用，构造相应的 SAL，需要什么样的系统调用，就构造什么样的抽象层。从而使 SAL 具有很大的伸缩性。

(1)伸缩性

为了保证 SAL 有很好的可扩展性，预编译开关的使用、数据类型的封装是必需的。其中预编译开关根据类型可分成操作系统预编译开关和功能特性预编译开关。

①预编译开关的设计

a 操作系统预编译开关：用于指示 SAL 对不同手机系统的支持，如：OS_MTK_6225、OS_MTK_6228、OS_ANYKA_T500、OS_COOLSAND 这类预编译开关的使用是互斥的，且只有在 SAL 内部使用。针对不同的操作系统平台和版本，SAL 可启动相应的操作系统预编译开关。

b 功能特性预编译开关：用于指示 SAL 的具有各种不同的功能特性，这类预编译开关的使用有的是互斥的，有的则不是互斥的，可以根据目标平台需要打开或关闭特定功能的预编译开关。

②数据类型的设计

为了保证软件的可移植性，一般不直接使用操作系统的原始数据类型，需要针对不同的平台将原始的数据类型进行封装。在 SAL 中，封装的基本数据类型为 VMUINT8/VMINT8(无符号 char 类型/有符号 char 类型)，VMUINT16/VMINT16(无符号 short 类型/有符号 short 类型)，VMINT32/VMUINT32(无符号 long 类型/有符号 long 类型)，VOID 等。

(2)系统事件

基于 VRE 的程序被加载后形成新的进程，进程创建后立即处于运行状态，当进程被外来事件中断后，进程转入暂停状态，外来事件被处理后进程恢复为运行状态，进程无论在暂停还是在运行状态都可以被停止。VRE 为应用程序提供系统消息接口，能够让应用程序得到状态迁移消息。

(3)输入接口

手机的输入接口包括键盘输入和触摸屏输入，不同型号手机会支持不同的输入方式。在集成键盘输入接口时，VRE 把不同平台的键盘码统一为标准键盘码，应用程序只需要处理 VRE 预定义的键盘码，无需关心异构平台的实现细节。集成触摸屏输入接口时，VRE 触控笔的按下、离开和拖动三种事件。

(4)内存操作

通过操作系统内存分配接口申请一块连续内存用作内存池供 VRE 使用。VRE 通过自有的内存管理模块，比如可以借鉴 Doug Lea 在 Unix 上的实现算法[7]来管理该内存池，当 VRE 退出时系统释放掉该内存池。

(5)图形设备

VRE 所需的最小图形设备接口是获得屏幕直写接口，当 VRE 获得屏幕直写能力后，通过集成的图形库可以向应用程序提供高级图形绘制功能。

如果目标平台具备高级图形函数，VRE 的高级图形函数可以优先封装平台 API，只当目标平台的高级图形函数缺失或未实现时才用内置图形算法替换，这是因为目标平台往往使用硬件实现图形绘制功能，在速度和内存开销上都优于软件实现。

(6)TCP/IP 协议栈

VRE 向应用程序提供类 BSD SOCKET 接口，使用异步通信模型，应用程序通过回调函数获知协议栈事件。

(7)时间接口

时间接口包括定时器和时钟。

为了降低目标平台负载，VRE 只需要目标平台提供一个定时器，然后用软件方式模拟出多个定时器供应用程序使用。由于 VRE 定时器是软件实现，而且是在单任务中模拟实现，因此 VRE 定时器不可重入。

(8)文件系统

文件系统接口构建在目标平台的 API 之上，即 VRE 不关心目标平台用那种文件系统，只要求目标平台提供必要的文件操作 API。

5 总结

在低端手机市场开发的成本、应用的丰富程度是

(下转第 113 页)

© 中国科学院软件研究所 http://www.c-s-a.org.cn

(上接第 87 页)

至关重要的因素。因此一个集成成本低、重用行高、扩展性好的中间件就显得非常必要。VRE 提供一个中间层，屏蔽底层手机平台，向上为应用程序提供一个统一的运行平台，提供一个统一简单的应用程序接口，避免了应用程序开发人员对底层操作系统的了解，支持快速有效的开发软件。VRE 的性能有：

(1)可移植性。应用程序通过标准规范的接口来实现，平台独立于下层具体的操作系统，屏蔽下层系统的实现细节，当根据需要选择不同的操作系统时，应用程序不用作或只用作极小的改动，即应用可以在不同的手机平台之间移植。

(2)缩短项目的开发周期。软件无需修改，就可方便的被更新和扩展。

(3)平台具有较好的可维护、可重用和伸缩性。丰富的功能模块和灵活的定制机制，并且具有规范的开发/扩展框架，使得平台开发者可以根据规范的设计模式对平台功能进行维护、扩展和继承。

参考文献

1 Fox D, Verhosek R. Micro Java Game Development. New York:Addison Wesley, 2002:140－197.

2 Rischpater R. Software Development for the QUALCOMM BREW Platform. Washington: Microsoft Press, 2003: 51－59.

3 Lee V, Schneider H, Schell R. Mobile Applications: Architecture, Design, and Development. Indiana: Prentice Hall PTR, 2004: 61－62.

4 王小旭.手机应用平台技术剖析.电子世界, 2007,(6).

5 Wells MJ. J2ME Game Programming Network: Muska and Lipman/Premier-Trade, 2004:30－42.

6 汤子瀛,哲凤屏,汤小丹.计算机操作系统.西安:西安电子科技大学出版社, 1996.

7 Wilson P, Johnstone M, Neely M. Dynamic Storage Allocation: A Survey and Critical Review. Proceedings of the 1995 International Workshop on Memory Management. Springer LNCS, 1995.

© 中国科学院软件研究所 http://www.c-s-a.org.cn