

华东师范大学
硕士学位论文
基于Java的移动终端中间件的研究与设计
姓名：刘向辉
申请学位级别：硕士
专业：软件工程
指导教师：黄国兴
20050401

论文题目：基于 Java 的移动终端中间件的研究与设计

摘 要

随着移动通信技术的飞速发展,移动业务也从最初的通话、短消息等基本服务向更加丰富和复杂的功能演进,如:移动浏览、游戏娱乐、移动商务、视频服务、位置服务等。移动通信运营商需要有一个移动终端运行和开发的统一平台来提供对越来越多的移动应用的支持,即移动终端中间件。

本课题是中国电信 3G 终端研究项目之一,由中国电信上海研发中心承担,作者参与了其中的研发工作。所谓移动终端中间件,不同于传统的中间件技术,它是位于移动终端底层硬件或操作系统平台之上,应用层之下的中间适配层。该层向下屏蔽掉硬件平台或操作系统平台的差异,向上为应用层提供一套统一的标准接口,应用层的开发基于该接口进行。

本文分析了 3GPP 移动终端的体系结构及其软硬件技术,总结了移动业务的开发现状以及移动业务开发中遇到的问题,提出了移动终端中间件的设计目标,总体架构和技术方案等。本文的重点是根据移动业务将移动终端中间件定义成较为完整的功能集组合,其中平台层定义了较为底层的系统级 API 集,为模块层提供支撑;模块层则定义了较为上层的基本功能集(SMS、MMS、WAP、多媒体等)和扩展功能集(蓝牙、SIP、位置服务、Web Services、加密和信任服务等),用户可以直接使用某些功能模块完成移动应用的客户端软件开发。另外,本文以 SMS、MMS 等为例给出了部分功能集的应用层实现方法,同时,就中间件的版本演进、平台无关性、安全策略和数字签名、应用管理软件、应用开发以及测试提出了建议的方案。最后,设计了基于 Java 的移动终端中间件的实现,为运营商提供了根据 JTWI 扩展的功能集,并通过一个实例介绍了基于移动终端中间件的业务开发方法。

移动终端中间件为无线应用开发提供了一个统一的软件平台,缩短了业务的客户端软件的开发周期,并增强可扩展性。移动终端中间件为实现无线数据应用的可移植性、可重用性和模块化提供了坚实的基础。

【关键词】中间件; 终端; Java

Title: A Study and Design for Middleware of Mobile Terminal Based on Java

Abstract

With the rapid development of mobile communication technology, mobile service are evolving from basic function, as calling and SMS, into abundant and complicated function, as mobile browsing, gaming, mobile commerce, video, LCS and so on. Mobile operator should have the Middleware of mobile terminal which could provide a generic platform for the mobile application.

This project is 3G terminal research of China Telcom, developed by STTRL. Middleware of mobile terminal is different from traditional middleware. It's an adapter layer over the hardware or operating system and below the application layer of terminal. This layer shield the difference between kinds of hardware and operating system platform, and provide standard interface to application layer.

This thesis analyses the architecture, hardware and software technology of 3GPP terminal, summarizes the actuality and problem of mobile service development. According to the requirement of services, the author designs the object of mobile terminal Middleware and the system architecture, technical solution. The emphasis of this thesis is the definition of the function sets according mobile services. The middleware is constitutive of platform layer and functional layer, and the former define the basic APIs of low level, and the latter define the basic functional sets as SMS, MMS, WAP, multimedia and extended functional sets as bluetooth, SIP, LCS, Web Services and so on. User can develop mobile application with these functional sets. At the same time, the paper advances the technique of version evolving, transplanting, security, application manager, application development and testing. At last, the author gives the realization of mobile terminal Middleware based on java, and describes the development method with an example.

Middleware of mobile terminal is a universal software platform for application development. The technology can simplify the development of client software, and improve the expansibility, importing, reuse and modularization of wireless data applications.

【Key Words】 Middleware, Terminal, Java

学位论文独创性声明

本人所呈交的学位论文是我在导师的指导下进行的研究工作及取得的研究成果。据我所知，除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名：刘向辉 日期：2005.5.25

学位论文授权使用声明

本人完全了解华东师范大学有关保留、使用学位论文的规定，学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版。有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆被查阅。有权将学位论文的内容编入有关数据库进行检索。有权将学位论文的标题和摘要汇编出版。保密的学位论文在解密后适用本规定。

日期：2005.5.25

学位论文作者签名：

刘向辉

日期：2005.5.25

导师签名：

郭元

第一章 绪 论

1.1 移动终端概述

20 世纪 90 年代以来,移动通信行业在市场和技术的驱动下发展迅速,经历了第一代移动通信(模拟)系统和第二代移动通信(数字)系统,现在正向第三代移动通信系统过渡发展。据最新的统计结果显示,中国移动通信市场用户数量已经超过 3.25 亿户,中国平均每 100 人中就有 25.1 人拥有一部手机,移动终端的巨大市场有目共睹。

即将来临的第三代移动网络和宽频技术,更带给了我们对于无线通讯未来的美好期望。3G 时代,移动终端不仅可以有更强大的通讯功能,还能提供其它丰富的附加应用功能,用户可以从移动网络上获得大量的信息、软件和服务,充分利用已有的硬件资源来扩展移动终端的功能。这种将数据信息处理能力与无线通讯工具融合的趋势正在 3G 终端中得到实现。从某种意义上来说,除专有的无线网络协议的处理功能外,3G 移动终端已经是一台小型化的、功能完整的、复杂的掌上电脑了。世界主流计算机技术已进入后 PC 时代。所谓后 PC 时代,即非 PC 信息设备大显神通的时代,以信息家电为代表的嵌入式系统正是非 PC 信息设备的主体,其应用广泛、领域特色突出,发展空间巨大。以信息设备为代表的移动互联网时代的嵌入式产品,不仅为嵌入式市场展现了美好前景,注入了新的生命;同时也对嵌入式系统技术,特别是软件技术提出新的挑战。这主要包括:支持日趋增长的功能密度、灵活的网络联接、轻便的移动应用和多媒体的信息处理等。随着通讯和计算技术的发展,嵌入式系统在生产和消费领域的应用将会越来越广泛,嵌入式系统技术的发展为移动通信终端的发展奠定了良好的技术基础。

移动终端已成为各种应用和业务的承载平台,因此对其硬件性能要求不断提高。3G 终端技术正逐步从单 CPU 构架向双 CPU 构架演进,通过专用的应用处理器满足日益增长的应用需求,终端开始以应用处理器为核心,移动通信模块只是它的一个外围电路,这也说明 3G 无线移动终端的功能逐渐不再以通话为主,而是具备各种我们日常需要的处理和通信功能。无线移动终端上的软件,主要包括嵌入式操作系统、应用运行平台、应用程序等。软件的复杂性最终由用户的需求决定,取决于用户对诸如照/摄像、MP3、收音机、蓝牙、红外等硬件配置的需求,以及 MMS、Java 应用、SyncML 应用、WAP 应用、多媒体等业务的需求。用户对业务的需求需要支撑技术的支持,支撑技术的运行需要操作系统的协调管理,操作系统的运作需要底层硬件强大的处理能力支撑。任何一个环节弱化都可能影响到用户的体验,最终影响移动业务的推广。

1.1.1 总体结构

3GPP 定义了 UMTS 系统的架构, 广义上移动终端在 UMTS 系统中是指用户设备部分 (User Equipment Domain, UE), 本文中的移动终端术语表示移动设备部分 (Mobile Equipment Domain, ME), 不包括 USIM 部分, 如图 1.1 所示:

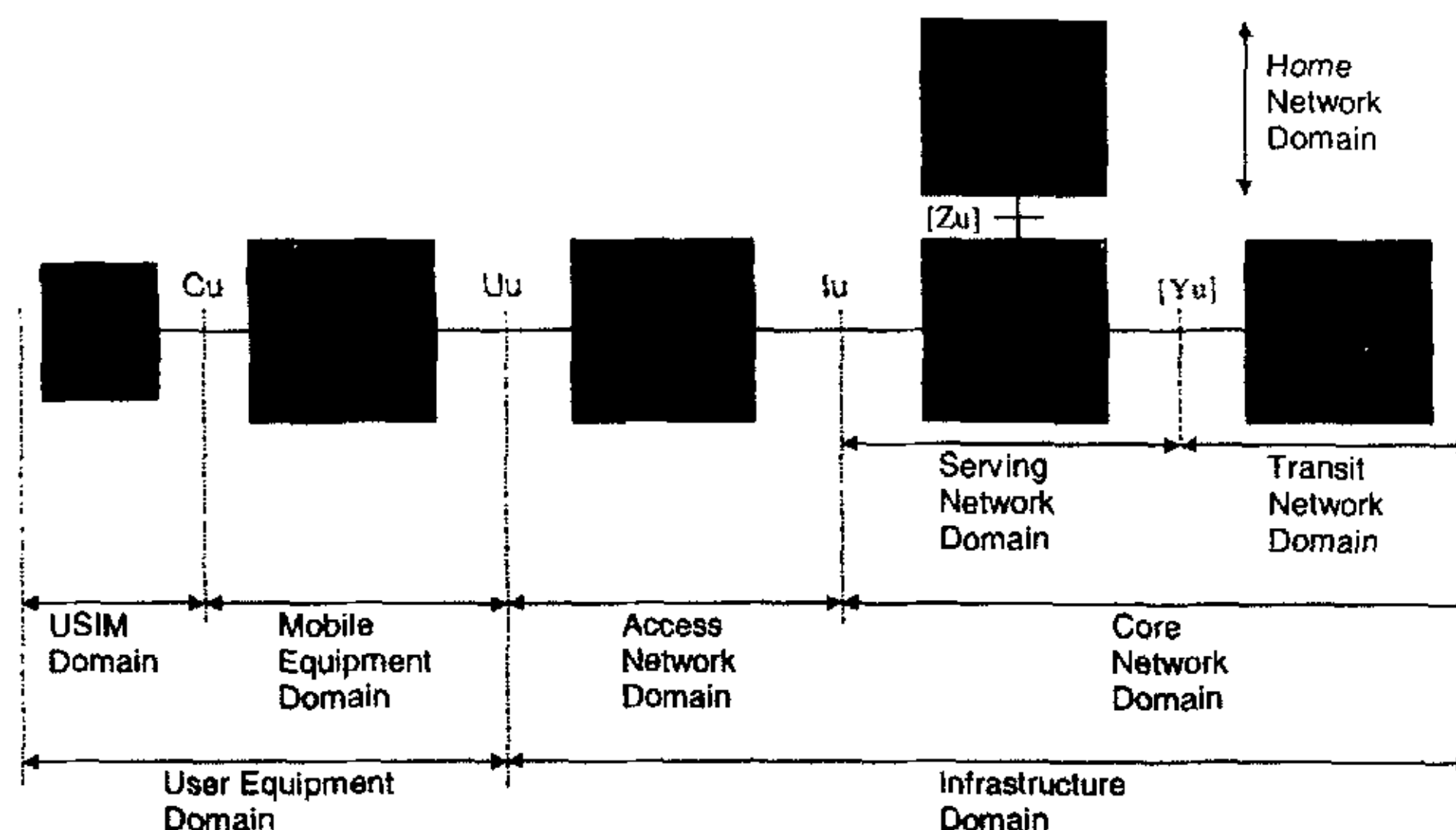


图 1.1 UMTS 系统^[1]

Cu = USIM 和 ME 之间的参考点;

Iu = 接入网和业务网之间的参考点;

Uu = 和基础架构域之间的参考点, UMTS 无线接口;

[Yu] = 业务网和传输网之间的参考点;

[Zu] = 业务网和归属网之间的参考点。

3GPP 将 UMTS 系统的终端概念化为 UE, 并在 TS23.101 中给出了相应的结构实现参考, 如图 1.2 所示。UE 根据机卡分离的原则, 分为 ME 和 USIM 两大部分。其中, ME 用于向用户提供无线接入与通信功能; 而 USIM 主要完成 UMTS 系统的用户识别和鉴权; 两者之间通过 Cu 接口连接。

与 2G/2.5G 不同, 3G 终端的参考模型中将 ME 的业务处理和无线接入功能划分为 TE 和 MT 两部分, 两者之间通过 R 接口相连。其中 MT 部分主要负责无线接入功能的实现, 而 TE 部分主要完成业务通信功能的实现。ME 通过 Cu 接口完成和 USIM 的数据交换和传输功能。USIM 基本功主要是存储数据和在安全条件下配合网络完成用户身份鉴权 and 用户信息加密、完整性算法的全过程。此外, 基于 USIM 提供增值业务正在逐步兴起, 例如利用其安全性以及动态菜单技术, 提供手机银行或移动支付服务等。

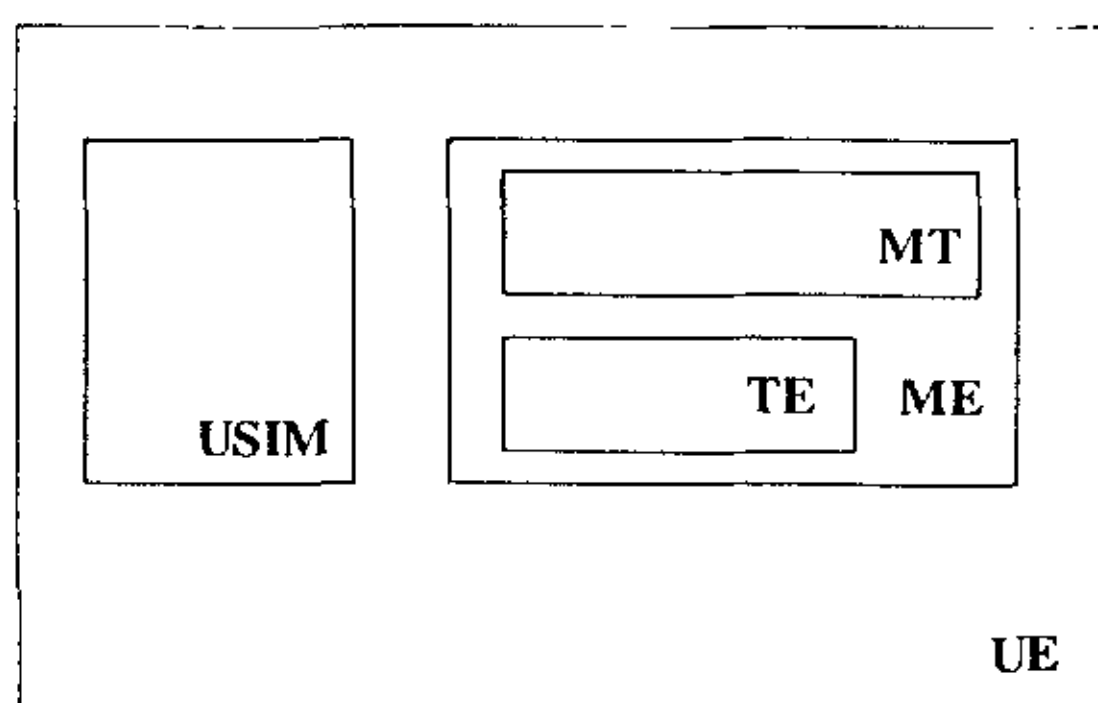


图 1.2 3GPP 关于 UE 的描述^[1]

1.1.2 硬件技术

3G 终端硬件平台的总体结构如图 1.3 所示，主要可以分为 4 大部分：RF 模块、通信协议处理与应用处理模块、功率控制模块、以及外围硬件功能模块等。其中，RF 模块、通信协议处理与应用处理模块、功率控制模块一般由芯片供应商以芯片组或套片形式提供，是终端硬件的核心技术，占终端成本的一半左右。而外围硬件功能模块一般由相应的第三方专业厂商提供，由终端厂商将其与核心芯片进行硬件集成。

终端解决方案的芯片组中，最重要的是通信协议处理与应用处理模块芯片或芯片组。该部分包括两个相对独立的功能模块：通信协议处理和应用处理。而典型 2G 语音业务为主的终端与 2.5G/3G 支持数据业务的终端之间，体系结构上最大的差异就在于这两部分功能模块是否单独存在。标准的 2G 终端上，典型的体系结构是没有应用处理器的单 CPU 结构。在该结构中，通信协议处理和应用处

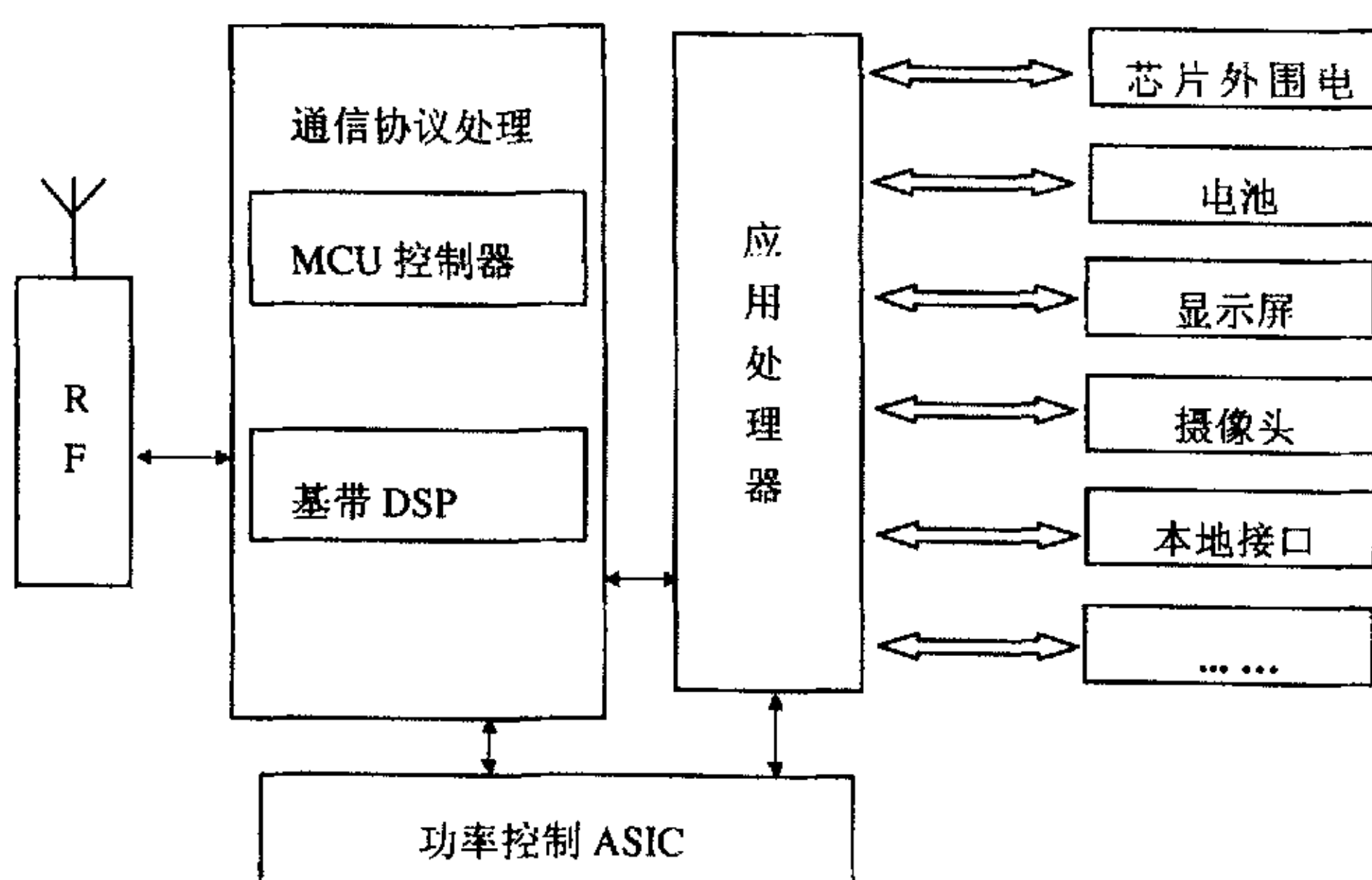


图 1.3 3G 终端硬件结构框图

理模块一般由集成在同一芯片中的 DSP 和 MCU 控制器组成。其中, DSP 主要用于实现无线调制解调和通信协议栈中物理层协议的功能, 也被称为基带 DSP。而 MCU 实际上一个基于 RISC 指令集的专用 CPU, 由一个运行其上的 RTOS 来进行控制, 并主要用来支持用户操作界面, 并实现通信协议栈中上层协议的各项功能。

随着数据的迅速发展, 2.5G 特别是 3G 终端在增加了通信带宽和协议复杂度的同时, 更大大加强了数据应用的处理需求。具有典型嵌入式系统特征的单 CPU 构架中的业务和应用处理功能被从 MCU 中分离出来, 由专用的应用处理器处理的双 CPU 混合式系统构架将逐步成为主流。采用双 CPU 构架后, 原嵌入式系统处理器将集中于更复杂的 3G 移动通信协议栈的处理, 并在剥离了无关的应用以及界面处理功能后, 能更好地进行优化以提高运行效率。此外, 由于该部分功能和技术都相对成熟稳定, 在终端研发中也无需频繁改动, 有利于加快产品开发进度。而应用处理器部分将负责应付各种现有和将来的不同应用, 其处理能力要求也将有很大差别, 将是终端产品中需要重点关注的部分。此外, 为了更好地适应用户界面以及应用处理的需要, 运行于应用处理器上的操作系统将更强调通用性能优化。

而对于终端所采用的 DSP 而言, 同样将需要更大的处理能力和更强的功能。DSP 最重要的功能将用于数字基带处理以及通信协议栈处理。此外, 终端的其他功能也都需要 DSP 资源的支持, 如语音识别、音频解压缩、图像/视频编解码等。系统对 DSP 芯片的 MIPS (每秒执行兆指令数) 消耗, 将从 2G 或 2.5G 手机的 50-100MIPS 增长到 500MIPS 以上。因此, DSP 芯片也可能与 CPU 类似产生分工, 未来手机很可能会采用两个或更多的 DSP 芯片——一个 DSP 芯片用来实现固定功能, 嵌入式调制解调器; 另外的 DSP 芯片用于灵活支持实现各种应用。

1.1.3 软件技术

正如前面所述, 智能终端大都采用双 CPU 结构, 围绕这两个 CPU 形成移动智能终端中的两个子系统: 通信子系统和应用子系统。其中通信子系统适应各种无线接口协议标准, 选择适当的移动通信网络, 建立和维持网络连接, 实现话音和数据通信; 应用子系统负责管理存储器、外围设备、外部接口等系统资源, 运行应用程序, 提供用户界面, 此外还包括终端的电源管理。

如此强大而复杂的硬件资源, 势必需要系统化管理, 因此单独的智能移动终端操作系统也就成为大势所趋, 它主要完成诸如进程、内存、外部设备等系统资源的调度和管理, 并为上层应用软件平台提供服务, 在操作系统之上可以执行各

种各样的应用程序。无线移动终端发展到了智能终端后，就产生了类似 PC 机的明确的逻辑结构。3G 终端软件属于嵌入式系统软件，它的最大特点就是软件固化在硬件中，软件代码必须保证高质量、高可靠性和实时性。图 1.4 描述了当前最为常见的 3G 终端软件结构，主要包括：第一部分是系统管理即增强型的操作系统，主要是控制各个功能模块对 CPU 中央处理器/内存等的硬件资源的使用，以及对电源进行管理。第二部分是 3G 网络通信协议，完成空中接口协议的传输。第三部分是驱动器，主要完成向应用软件及中间件提供为了控制硬件所需要的基本的接口和控制功能。第四部分是中间件，向应用软件提供显示、操作接口、通信协议处理、多媒体处理等共通功能的功能群。第五部分是上层应用软件，向各种向用户提供各种显示和操作功能。

目前市场上主流的终端操作系统主要有 Symbian、Windows CE、Palm OS、Linux 四种。移动终端上的应用运行平台主要包括 BREW 和 Java，作为移动通信数据业务平台，Java 和 BREW 的推出，标志着移动数据业务的开发与应用进入了新阶段。中国移动采用 Java 开发应用，而中国联通在选择 BREW 之后，又在市场的驱动下增加了 Java 应用。此外，韩国无线 Internet 标准化论坛(KWISF)为了摆脱高通的垄断，开发出基于 Java 的无线 Internet 操作平台——WIPI。KWISF 是韩国网络运营商 SK Telecom、KTF 和 LG TeleCom 三家公司的一个联盟组织，他们借助运营商在移动通信产业链中的核心地位，共同将 WIPI 在韩国推向商用，这也是终端技术标准化的一个例子。

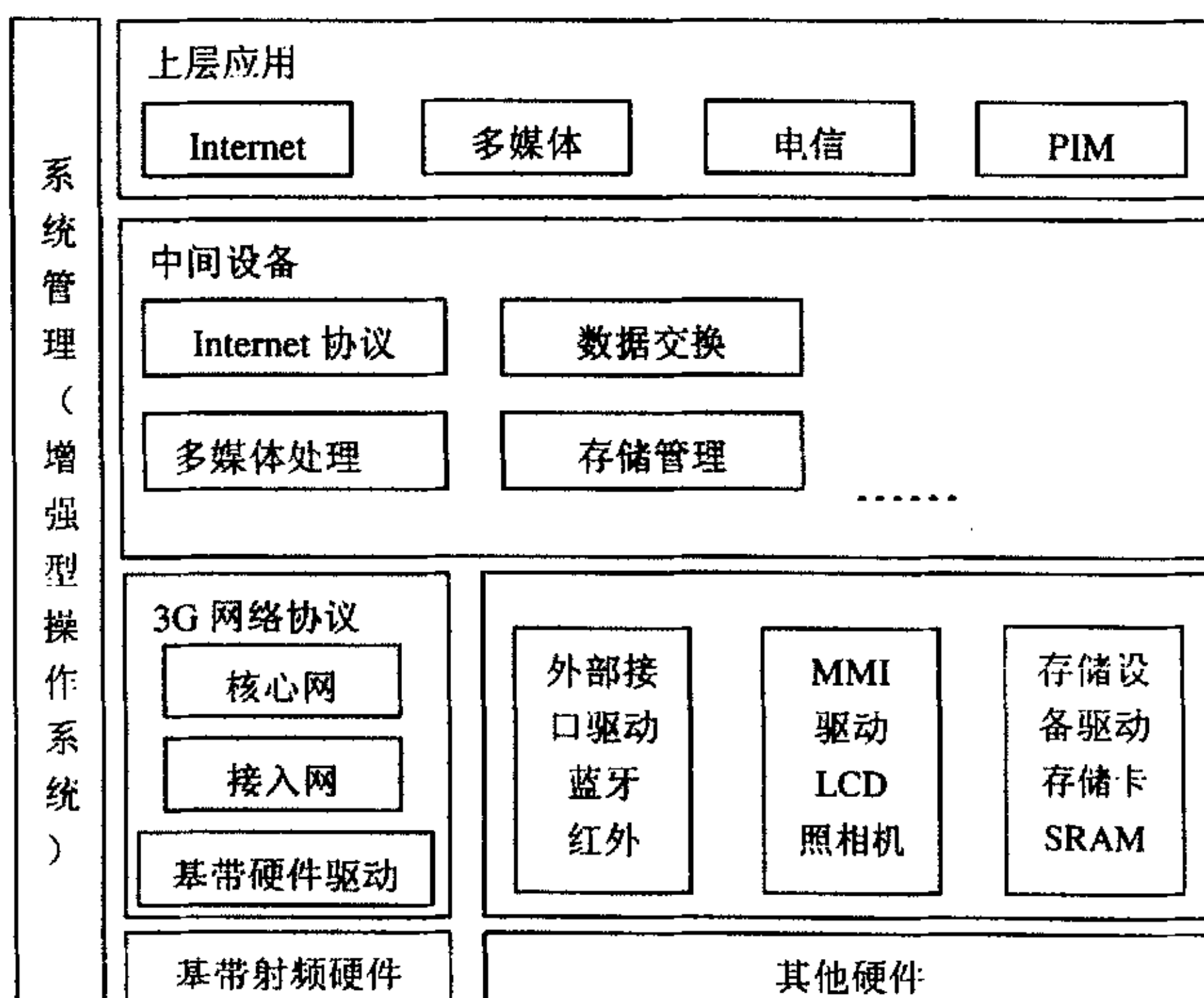


图 1.4 3G 终端软件结构

从移动业务的角度看,业务需要多种技术的支撑,比如移动互联网的 WAP 技术,个人信息管理(PIM)应用的 SyncML 技术,在线视频的流媒体技术等等。操作系统和应用运行平台只是业务运营的一个基础平台,要使业务运行起来还必须这些支撑技术的支持。

1.2 移动业务开发概述

1.2.1 移动业务开发现状

目前,移动通信业务主要集中在传统的语音通话业务和短消息业务,通话和短消息占据了移动通信业务的绝大部分,同时,基于 GPRS 或者 CDMA 的数据业务正在悄然增加。概括来讲,国内主要移动通信业务列举如下:

- 语音业务: 传统通话业务、其他增值语音业务;
- 消息业务: SMS、EMS、MMS、Email;
- 浏览业务: 移动互联网业务;
- 下载业务: 图铃、应用下载、离线应用、在线应用;
- 定位业务: 定位以及相关 GIS 业务;
- 宽带业务: 接入以及其他增值业务;
- 卡类业务: 基于 STK 智能卡的 IOD 等业务;
- 其他业务: 其他终端业务,如 PIM、MP3、BlueTooth 等业务。

由于当前的移动业务开发的无序性,作为产业链中的重要组成部分—移动运营商必然希望业务能够快速的推向市场,未来的移动通信产业的开发模式必然以移动运营商为产业链的主导,如图 1.5 所示:

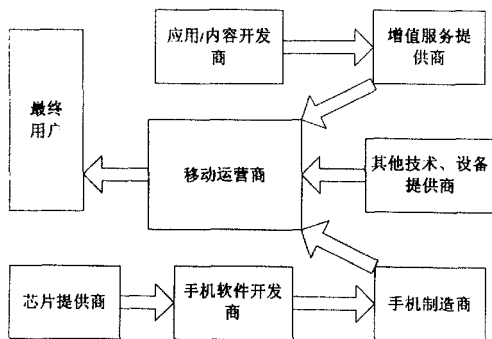


图 1.5 以运行商为中心的产业链

1.2.2 移动业务开发中所遇到的问题

无线数据行业的现状与个人电脑行业 20 年前的情形非常相似。由于不同的电脑品牌和款式间没有一个通用的应用软件平台,程序员只能为特定设备编写程序,而消费者也被局限在狭小的范围内作选择。但是通用个人电脑平台的出现使开发者可以为包括几乎所有计算机的大众市场编写代码,导致新软件的数量呈爆炸性增长,软件产品的普及推动了硬件的销售,消费者可以选用他们喜欢的应用来安装到自己电脑中,通过升级系统来更好地发挥新软件的性能。

对于无线网络运营商来说,通往赢利的道路总是意味着需要为其用户提供有价值、并且在一定程度上使之欲罢不能的服务。至今为止,语音业务一直是无线网络中的主打应用,但显而易见的是,非语音业务(数据应用或增值业务)也蕴藏着无限商机,如果运营商运作得当,并能摸准大众市场的喜好,这一市场必将迸发出无限的活力,这些数据应用包括实时信息服务、游戏、新闻、娱乐以及效率提升工具等,此处不一一列举。除此之外,这些应用还包括那些能够使运营商更加高效的与用户交流并降低用户流失率的应用,比如用以定制用户界面的应用以及能够更新基于手机功能的网络特性的应用,同时,新型基于语音的应用,如语音识别功能、一键通话等。

不断更新的业务给各移动终端厂商、应用提供商的开发带来了诸多问题,而且不同厂商的实现方式不同,很难保证业务的互通性和跨平台性;对于移动通信运营商来讲,更是延长了新业务的推出时间;同时,运营商很难控制和规范业务的核心技术,不能保证对最终用户的服务质量。无线行业向用户提供这些应用的最佳途径就是利用这样一种解决方案:开放、统一标准、独立于硬件,并且可以连续应用于任何网络、任何移动设备;提供端到端的解决方案,同时采用开发的技术平台以及完善的商业模式,能够在整个价值链上实现收入共享;能够使用户通过其无线设备发现、购买、下载并管理应用。对于移动运营商来讲,业务开发中的核心问题是终端缺乏一个统一、开放的无线数据应用平台,即移动终端中间件技术。

1.2.3 什么是移动终端中间件

针对移动终端的中间件技术正是为了解决上述这些问题而应运而生的。中间件是位于平台(硬件或操作系统)和应用之间的通用软件,具有标准的程序接口和协议。针对不同的操作系统和硬件平台,它们可以有符合接口和协议规范的多种实现。也就是说,中间件是位于底层硬件/操作系统平台之上,应用层之下的中间适配层。该层向下屏蔽掉硬件平台或操作系统平台的差异,向上为应用层提

供一个统一的标准接口，应用层的开发基于该接口进行。

目前，数字电视中间件技术已经成熟：MHP（基于 Java）已经成为 ETSI 标准，西欧的数字电视发展基本没有碰到业务兼容性问题。移动终端中间件目前尚无统一的标准，可以借鉴数字电视中间件技术思想开发适合于移动终端的中间件技术。

使用移动终端中间件技术可以：缩短应用的开发周期；节约应用的开发成本；减少系统初期的建设成本；降低应用开发的失败率；保护已有的投资；简化应用集成；减少维护费用；提高应用的开发质量；保证技术进步的连续性；增强应用的生命力等。使用中间件技术必将为移动终端带来突破性的发展，并且对于整个移动通信产业链产生巨大影响。

1.3 本论文的研究背景和主要贡献

1.3.1 本论文的研究背景

本研究课题是中国电信 3G 终端研究项目之一，由中国电信上海研发中心移动通信部终端项目组承担。在研究了大量的移动终端软硬件技术、从运营商的角度分析了各种 3G 业务、制定了各种 3G 业务的企业标准之后，运营商认为移动终端中间件技术对未来移动通信产业将产生重大影响。运营商需要掌控移动终端中间件技术，才能快速推出各类业务，提高用户的 ARPU 值，同时带动终端厂商、软件提供商、CP/SP 等，从而盘活整个产业链。

本项目启动之前，我们已经做了如下的研究工作和技术积累，包括：J2ME、BREW、MHP、Symbian、S60、WinCE 等技术的研究；制定了中国电信 3G 终端 Java 业务规范和测试规范，制定中国电信 3G 终端定制规范。另外，中国电信上海研发中心已经具备如下的实验环境：WCDMA 试验网络、下载引擎（支持 HTTP 下载、OMA 下载、MIDP 载）、3G 终端（支持 Java）、各类基于 Java 的开发环境和工具等。

中国电信移动终端中间件技术的项目计划如下：

- 1) 确定移动终端中间件技术方案和技术体制
- 2) 制定中国电信移动终端中间件功能规范
- 3) 制定中国电信移动终端中间件实现规范
- 4) 研发中国电信中间件参考设计
- 5) 终端厂商进行集成

- 6) 应用提供商介入, 开发应用
- 7) 推出中国电信的定制终端, 提供移动服务

1.3.2 本论文的主要贡献

本论文的研究重点主要集中在中国电信移动终端中间件项目的第二和第三阶段, 即中国电信中间件功能规范和中间件实现规范的研究和制定。本论文的研究目标是要分析当前各种移动终端中间件技术, 设计符合移动运营业务开发需求的中间件技术方案。本论文的主要贡献包括:

分析了 3GPP 的移动终端体系结构以及软硬件技术, 总结了移动业务的开发现状以及移动业务开发中遇到的问题, 提出了移动终端中间件技术的概念以及运营商主导的移动业务开发模式。

总结了传统中间件技术, 分析了移动终端中间件技术和传统中间件技术的联系和区别, 分析了移动终端中间件技术的特点和所涉及的关键技术: 中间件和移动终端平台技术、Java 虚拟机技术等, 为运营商设计符合产业链需要的移动终端中间件提供了技术依据。

深入分析了移动业务的需求, 提出了移动终端中间件的设计目标, 提出了移动终端中间件的总体架构和技术方案。本论文的重点是根据业务划分定义了较为完整的功能集组合, 并以短消息 (SMS)、多媒体消息 (MMS) 等给出了部分功能集的应用层实现。另外, 就中间件的版本演进、移植、安全、应用管理、应用开发以及测试提出了建议的方案。

提出了基于 Java 的移动终端中间件的实现, 为运营商设计了根据 JTWI 扩展的功能集, 为移动终端中间件的实现奠定了良好的基础。另外, 通过一个实例的开发介绍了基于移动终端中间件的业务开发模式。

第二章 关键技术研究

2.1 中间件技术现状分析

在中间件产生以前, 应用软件直接使用操作系统、网络协议和数据库等开发, 开发者不得不直接面对许多复杂棘手的问题, 如操作系统的多样性、繁杂的网络程序设计和管理工作、复杂多变的网络环境、数据分散处理带来的不一致性问题等等。这些与用户的业务没有直接关系, 但又必须解决, 耗费了大量的时间和精力。于是, 有人提出将应用软件所面临的共性问题进行提炼、抽象, 在操作系统之上再形成一个可复用的部分, 供应用软件重复使用。这一技术思想的发展最终诞生了中间件这一新的软件形态。中间件在操作系统、网络和数据库之上, 应用软件的下层, 总的作用是为处于自己上层的应用软件提供运行与开发环境, 帮助用户灵活、高效地开发和集成复杂的应用软件。用户在使用中间件时, 往往把一组中间件集成在一起, 构成一个平台(包括开发平台和运行平台), 在这组中间件中又必须要有一个通信中间件, 因此可以比较形象地把中间件定义为: 平台加通信。在具体实现上, 中间件是一个用 API 定义的分布式软件管理框架, 具有强大的通信能力和良好的可扩展性。

2.1.1 传统中间件技术

随着计算机软件技术的发展, 中间件技术也已日渐成熟, 并且出现了不同层次、不同类型的中间件产品。根据中间件在系统中所起的作用和采用的不同技术, 大致划分为以下五类。

(1) 数据库中间件

数据库中间件(Database Middleware, MD)在所有的中间件中是应用最广泛、技术最成熟的一种。它允许应用程序和本地或者异地的数据库进行通信, 并提供了一系列的应用程序接口 API。当然, 在多数情况下这些 API 都是隐藏在开发工具中, 不被程序员直接使用。最典型的例子是: JDBC、ODBC, 它们都是基于数据库的中间件标准。不过在数据库中间件处理模型中, 数据库是信息存储的核心单元, 中间件完成通信的功能。这种方式虽然是灵活的, 但是并不适合于一些要求高性能处理的场合。因为它需要大量的数据通信, 而且当网络发生故障时, 系统将不能正常工作。可见, 系统的灵活性提高是以处理性能的降低为代价的。

(2) 远程过程调用中间件

远程过程调用中间件(Remote Procedure Call, PRC)沿用了大多数程序员都

非常熟悉的编程模式——客户端/服务器模式。程序员就像调用本地过程一样在程序中调用远程过程。启动远程过程的运行，然后将运行结果返回给本地程序。不仅如此，远过程调用还可以将程序的控制传递到远端的服务器中。RPC 的灵活特性使得它有比数据库中间件更广泛的应用，可以应用在更复杂的客户端/服务器计算环境中。远程过程调用的灵活性还体现在它的跨平台性上，这种调用是可以跨不同操作系统平台的，而程序员在编程时并不需要考虑这些细节。RPC 也有一些缺点，主要是因为 RPC 一般用于应用程序之间的通信，而且采用的是同步通信方式，因此对于比较小型的简单应用还是比较适合的。但是对于一些大型且要求异步通信的应用来讲，这种方式不是很适合。因为此时程序员需要考虑网络或者系统故障，处理并发操作、缓冲、流量控制以及进程同步等一系列复杂问题。

(3) 面向消息中间件

消息中间件 (Message Oriented Middleware, MOM) 提供了一个完整的处理环境，允许开发者及用户连接不同系统之间的数据和代码，或采用一致的界面进行应用处理的互连。它提供了一个高层应用接口，为不同系统提供操作核心。消息中间件的工作主要是通过将信息以消息的方式在程序间传递来完成。消息中间件的优点主要是能够在客户和服务器之间提供同步和异步的连接，并且在任何时刻都可以将消息进行传送或者存储转发，这也是它比远程过程调用更具优势的原因。另外消息中间件不会占用大量的网络带宽，可以跟踪事务，并且通过将事务存储到磁盘上可实现网络故障时的系统恢复。当然和远程过程调用相比，消息中间件不支持程序控制的传递，不过这种功能和它的优势比起来是无关紧要的。消息中间件适用于需要在多个进程之间进行可靠数据传送的分布式环境。

(4) 基于对象请求代理的中间件

基于对象请求代理 (Object Request Broker, ORB) 的中间件主要采用面向对象的技术，它可以看作是与编程语言无关的面向对象的 RPC 应用。ORB 被视为从面向对象过渡到分布式计算的强大推动力量。从管理和封装的模式看，对象请求代理和远过程调用有些类似，但对象请求代理可以包含比远过程调用和消息中间件更复杂的信息，并且可以适用于非结构化的或者非关系型的数据。目前，ORB 存在二个彼此竞争的标准：CORBAORB 和 DCOMORB。当使用 ORB 时，IDL (Interface Define Language) 用于定义对象之间的接口。它类似于 RPC 中的 IDL 定义过程的接口。ORB 特别适用于对象接口变化不频繁，不会导致代码经常被重新编译及链接的情况。

(5) 事务处理中间件

事务处理中间件 (Transaction Processing Monitor, TPM) 是一种复杂的中间件产品, 是针对复杂环境下分布式应用的速度和可靠性要求而实现的。它给程序员提供了一个事务处理的 API。程序员可以使用这个程序接口编写高速而且可靠的分布式应用程序, 即基于事务处理的应用程序。事务处理中间件向用户提供一系列的服务, 如应用管理、控制管理以及应用程序间的消息传递等。常见的功能包括全局事务协调、事务的分布式二段提交、资源管理器支持、故障恢复、高可靠性、网络负载平衡等。

2.1.2 移动终端中间件与传统中间件的区别

在服务器/应用服务器/客户机三层模式中, 中间件 (应用服务器) 是应分布式网络发展的要求而产生的, 突破了服务器/客户机的两层模式, 使应用软件独立于计算机硬件和操作系统。中间件应用设计应该是从异构的计算资源中创建一个“虚拟主机”。如图 2.1 所示, 应用服务器处于服务器和客户机之间, 专门负责处理服务器端的业务逻辑、管理分布式应用以及控制网络流量, 从而缩短服务器和客户机的应用开发时间。中间件在对事务完整性的保证、对大规模并发处理的响应、对异构系统互联的透明支持以及对应用数据的安全性保护等方面的表现将成为应用系统成败的决定性因素。可见, 三层体系结构中的中间件实质上是一种负责专门功能的服务器。

然而, 在移动终端中, 中间件是指居于终端内部操作系统与应用程序之间的软件模块, 如图 2.2 所示, 它以应用程序接口 API 的形式存在, 整个 API 集合被存储在终端中。从某种角度来说, 它与服务器/应用服务器/客户机三层模式下的中间件概念有很大不同: 移动终端中间件很大程度上指的是一种处于应用程序和底层硬件或底层操作系统之间的软件层。

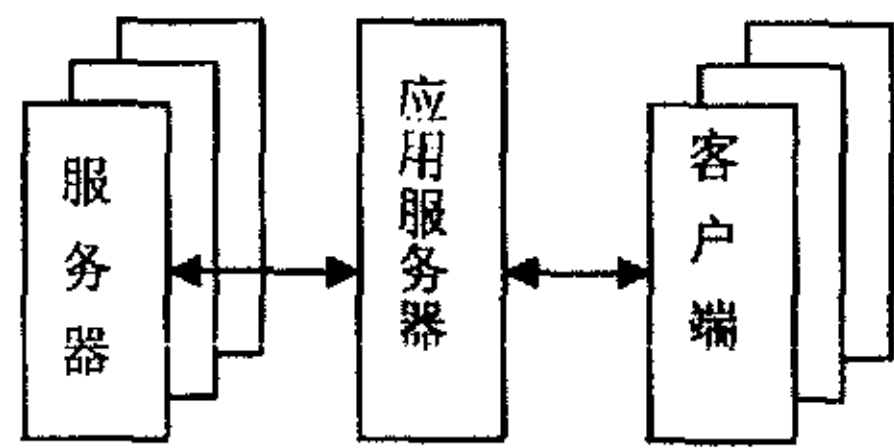


图 2.1 服务器/应用服务器/客户端结构

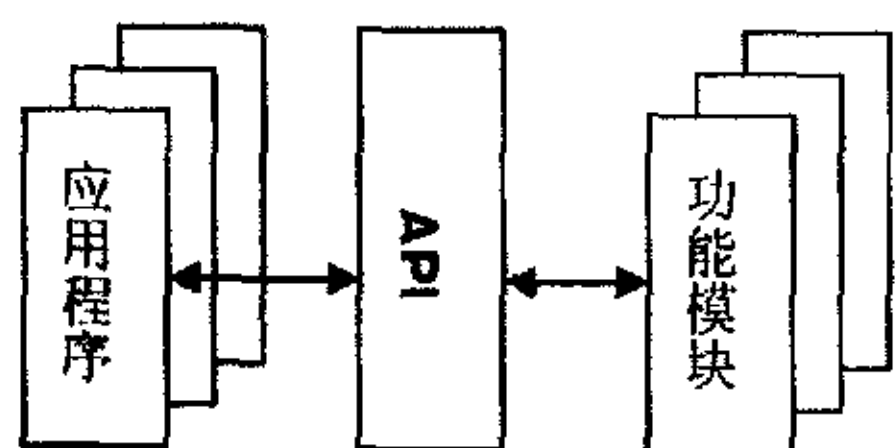


图 2.2 移动终端中间件结构

应用程序实现其功能的关键就是调用终端中的中间件所包含的 API, 即应用程序与各功能模块之间的接口。再由机顶盒执行应用程序, 完成用户请求的功能。因此, 中间件在终端中处于核心地位。中间件的目的是使终端通用的功能以应用

程序接口 API 的形式提供给终端生产厂家, 以实现移动业务功能的标准化, 同时使需要下载的服务项目 (以应用程序的形式存在) 所需的数据量减小到最低限度。中间件产品由运营商制定功能规范, 由除内容提供商和终端厂商之外的第三方提供, 这对移动业务的开发和终端的设计制造等过程的进一步简化和标准化都是非常有利的。

2.2 移动终端平台

当前主流的移动业务开发方法, 包括基于操作系统的业务开发方法、芯片级的解决方案、独立实现的解决方案等。本节根据运营商业务需求综合分析了各种技术方案以及移动终端中间件如何与这些技术融合。

2.2.1 操作系统级方案

目前, 大部分的移动终端应用都是操作系统相关的 (如 1.1.3 节描述的软件结构), 运营商可以逐步从这些操作系统和应用层中分离中间件, 这类的代表主要包括: Symbian、WinCE、Linux、Palm 等。其优点是: 功能集覆盖主要终端功能、潜在开发队伍规模较大、厂商自行发展; 其缺点是: 开放性较差、运营商无法掌握终端的业务功能。

2.2.2 芯片级方案

CDMA 技术主要由高通公司掌握, 其基于芯片的解决方案性能很高, 但是处于垄断地位, 运营商以及产业链的发展会受到制约。这类方案, 芯片厂商将解决方案软件进一步扩展实现, 代表有: Brew、APOXI 等。其优点是: 硬件级别的运行效率最高、厂商自行发展; 其缺点是: 硬件关联度大, 开放性最差、运营商无法掌握终端业务功能的发布、潜在开发队伍规模较小。

2.2.3 独立实现方案

独立实现的解决方案是在移动终端置入应用运行的平台, 应用可以自由安装和删除。运营商倾向采用与底层关联度较小的技术来实现。这类方案的代表有: J2ME、Flash 等。其优点是: 硬件关联度最小, 开放性最好、运营商可以掌握终端的业务功能发展、潜在开发队伍规模最大。其缺点是: 运行效率较差、开放性造成标准混乱, 私有扩展 API 泛滥, 需要由运营商统一。

2.3 虚拟机技术

2.3.1 Java 技术与 Java 虚拟机

Java 由四方面组成: Java 编程语言、Java 类文件格式、Java 虚拟机和 Java 应用程序接口(Java API)。它们的关系如下图所示:

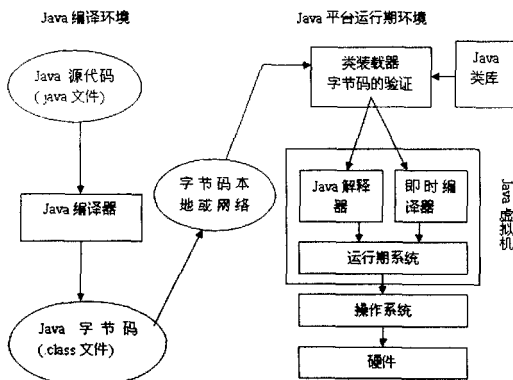


图 2.3 Java 四个方面的关系

运行期环境代表着 Java 平台，开发人员编写 Java 代码(.java 文件)，然后将之编译成字节码(.class 文件)。最后字节码被装入内存，一旦字节码进入虚拟机，它就会被解释器解释执行，或者是被即时代码发生器有选择的转换成机器码执行。从上图也可以看出 Java 平台由 Java 虚拟机和 Java 应用程序接口搭建，Java 语言则是进入这个平台的通道，用 Java 语言编写并编译的程序可以运行在这个平台上。这个平台的结构如下图所示：

在 Java 平台的结构中，可以看出，Java 虚拟机(JVM) 处在核心的位置，是程序与底层操作系统和硬件无关的关键。它的下方是移植接口，移植接口由两部分组成：适配器和 Java 操作系统，其中依赖于平台的部分称为适配器；JVM 通过移植接口在具体的平台和操作系统上实现；在 JVM 的上方是 Java 的基本类库和扩展类库以及它们的 API，利用 Java API 编写的应用程序(application) 和小程序(Java applet) 可以在任何 Java 平台上运行而无需考虑底层平台，就是因为有 Java 虚拟机(JVM)实现了程序与操作系统的分离，从而实现了 Java 的平台无关性。

JVM 在它的生存周期中有一个明确的任务，那就是运行 Java 程序，因此当

Java 程序启动的时候，就产生 JVM 的一个实例；当程序运行结束的时候，该实例也同时消失。

2.3.2 Java 虚拟机的体系结构

JVM 可以由不同的厂商来实现，由于厂商的不同必然导致 JVM 在实现上的一些不同，然而 JVM 还是可以实现跨平台的特性，这就要归功于设计 JVM 时的体系结构了。

一个 JVM 实例的行为涉及到 JVM 的子系统、存储区域、数据类型和指令这些部分，它们描述了 JVM 的一个抽象的内部体系结构，其目的不仅规定实现 JVM 时它内部的体系结构，更重要的是提供了一种方式，用于严格定义实现时的外部行为。每个 JVM 都有两种机制，一个是装载具有合适名称的类（类或是接口），叫做类装载子系统；另外的一个负责执行包含在已装载的类或接口中的指令，叫做运行引擎。每个 JVM 又包括方法区、堆、Java 栈、程序计数器和本地方法栈这五个部分，这几个部分和类装载机制与运行引擎机制一起组成的体系结构图

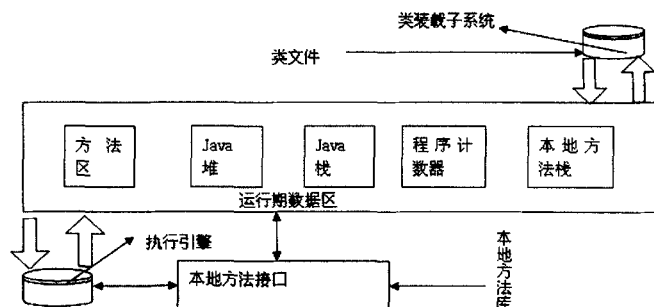


图 2.4 JVM 的体系结构

JVM 的每个实例都有一个它自己的方法域和一个堆，运行于 JVM 内的所有的线程都共享这些区域；当虚拟机装载类文件的时候，它解析其中的二进制数据所包含的类信息，并把它们放到方法域中；当程序运行的时候，JVM 把程序初始化的所有对象置于堆上；而每个线程创建的时候，都会拥有自己的程序计数器和 Java 栈，其中程序计数器中的值指向下一条即将被执行的指令，线程的 Java 栈则存储为该线程调用 Java 方法的状态；本地方法调用的状态被存储在本地方

法栈，该方法栈依赖于具体的实现。

2.4 小结

本章介绍了现有中间件技术，分析了移动终端中间件技术和传统中间件技术的联系和区别，总结了移动终端中间件技术的特点和所涉及的关键技术：中间件和移动终端平台技术、Java 虚拟机技术等。运营商可以通过整合现有技术，设计符合产业链需要的移动终端中间件。

第三章 移动终端中间件技术

3.1 需求分析

移动终端中间件技术要满足移动通信运营商业务开发和运营的需求，同时应该兼顾产业链上的终端设备厂商、软件厂商以及 CP/SP 的技术和利益。

移动终端中间件技术的实现需要有较全的产品线，支持多种 CPU 和嵌入式操作系统；运行效率要相对较高；可以实现根据中高低端产品实现裁减和配置，需要有良好的灵活性、可移植性和可伸缩性的特点，采用层次化、模块化的设计思想；具有良好的 API 接口设计，为移动业务的客户端软件开发提供优秀的运行和开发环境。

3.1.1 移动业务划分

由于移动业务越来越趋向多样性和复杂性，根据不同业务功能，有必要将当前和未来的移动业务进行系统的分析并且划分，同时，根据业务划分产生对应的功能接口集，如下表所示：

表 3.1 移动业务划分

业务分类	功能模块	详细说明
电话应用	电话特性	呼叫创建、来电呼叫、呼叫激活、通话计时、通话计费通知、禁止呼叫、呼叫等待、呼叫应答、呼叫拒绝、呼叫转接、三方通话、呼叫切换、静音/非静音、发送预定义的 DTMF 铃音、呼叫转移、紧急呼叫支持以及其他特性
	电话服务和特性	录音器、电话设置、声音拨号以及声音标记
	快捷拨号	对应键盘的按键自定义分配电话号码
	通讯记录	显示姓名、号码、呼叫指示以及图片
	消息指示	显示已接电话、已拨电话、未接电话、呼叫费用以及呼叫后的呼叫信息概要
	情景模式	设置名称、铃音、音量、声音类型、分组、个性化铃音、震动、消息接收、按键、通知和警告铃音。情景模式可以被改变、个性化配置以及重新命名
	增值语音业务	语音信箱、VOIP、SIP、POC 等
PIM	名片夹	联系人数据库，整合了消息和其他应用，并支持 vCard。
	日历	行程安排应用，支持 vCalendar 数据格式
	待办事项列表	vCalendar 项可被下载到待办事项列表中
	记事本	创建、修改、删除记事文本
	图象管理	存储、编辑多媒体消息和图象

	同步	SyncML 同步引擎。支持基于移动网络 (WAP)、蓝牙、红外 (IrDA) 的数据传输。支持 vCal 以及 vCard 数据格式
	智能卡操作	SIM 卡中的 ADN、SMS 等管理和应用
浏览下载	WAP 浏览器	WAP 浏览器, 支持 WWW、Push、WTAI 和安全增强特性
	Client Provisioning	客户端参数自动配置
	HTTP 下载	标准 HTTP 下载, 可以下载任何文件
	Java 下载	标准 MIDP 下载, 可以下载 MIDlet
	OMA 下载	标准 OMA 下载, 可以下载任何文件
消息	SMS	标准 SMS
	MMS	标准 MMS
	其他	EMS、Email、IMPS
多媒体	图片	各类图片格式的支持
	铃音	各类音频格式的支持
	视频	各类流媒体格式的支持
	游戏	各类游戏和相关软件
	Camera	照/摄像
	3D	3D 图形应用
	矢量图	Scalable Vector Graphics 应用
	Flash	Flash 菜单、内容、工具、游戏等应用
用户界面	手机电视	数字地面广播相结合的移动业务
	UI	各种 UI 控件、Skin、Themes 的支持
常用工具	时钟	时钟、世界时钟等
	计算器	计算器、商业计算器等
	换算器	单位换算器等
	其他工具	各种实用的工具软件
安全性	安全策略	安全设置以及软件认证管理
	DRM	数字版权管理支持
	加密	支持各种加密算法
本地连接	OBEX 引擎	支持 OBEX 协议, 实现在不同的设备、不同的平台之间方便、高效的交换信息
	PC-connectivity	PC 连接以及管理功能, 需要结合 PC 实现相应的客户端软件交互
	蓝牙	支持蓝牙相关协议
	红外	支持红外相关协议
	SyncML 同步	支持 SyncML 协议
位置服务	LCS 服务	获得当前地理位置、终端定位、终端存储路标的访问等
	GIS 服务	基于定位的地理信息应用服务
其他	文件管理	文件以及 I/O 操作等
	资源管理	设备资源管理、电源管理等
	BREW	通过改造运营商门户和 WAP 浏览器实现对

		BREW 业务的支持
	Web Service	电子商务、企业应用等
	行业应用	其他针对专门行业开发的功能模块

3.1.2 系统设备要求

根据中国电信目前的业务要求,移动终端中间件将主要运行在 3G 终端上,移动终端中间件移植到目标设备上所需要的硬件配置列表如表 3.X 所示:

表 3.2 移动终端中间件硬件要求

项目	要求
显示	至少 176 x 208 象素的彩色显示
输入	双功能键、五向导航、应用启动和切换键,还有电话拨出和挂断键。为了更好更方便地输入文本,还包括了清除键以及一个编辑键。使用 12 个标准的标有字母字符的数字键盘
处理器	建议目标设备使用 32 位 ARM 处理器
ROM	视版本和所支持的功能集而定
RAM	视版本和所支持的功能集而定

同时,移动终端中间件需要实时操作系统的支持,为移动中间件提供底层的与硬件之间的交互和管理操作。

3.2 移动终端中间件的目标

智能电话类设备的成功在很大程度上依赖于第三方的创新性应用以及内容提供—即移动服务和商业应用的发展。最终用户正开始习惯向他们的设备下载内容和应用。因此终端的潜在价值将受高质量的服务、应用以及内容的影响。

手机设计和功能的多样化在大大增强,造成的结果是在相互竞争的设备中,它们的屏幕尺寸、键盘、浏览器及其他用户界面元素的类似之处很少。应用、服务及其他内容必须适应这些不同的设备。从短消息到多媒体,从 Sockets 到 WAP 浏览器,基于移动终端中间件的开发者可以拥有一个完整的通信领域 API(中间件提供的 SDK API 中将有详细的使用指南)。这些 API 不但使开发出的产品能提供给移动电话用户真正有用的特性,而且使开发者能够为基于数据的移动电话写出更优秀的代码和引人注目的应用程序。另外,对开放的通信标准的支持使他更易于应用和服务程序的开发。

移动终端中间件的目标就是将设备差别限制到一些诸如用户需求类的关键

因素。运营商可以通过移动终端中间件的授权,帮助手机制造商向市场推出与此平台功能相兼容的手机。实现标准化的应用环境有利于服务的创建以及应用间的交互。规范屏幕尺寸、输入方法、各种 API、用户界面库以及获支持的技术使得服务与应用可以进行无缝交互,但被授权的手机制造商仍被赋予革新以及设计优秀智能电话的自由。市场通过一个公共平台接入降低开发成本,因而有更广阔的应用范围,解决终端需求,创建良性循环。

3.3 总体结构和软件架构

3.3.1 总体结构模型

众所周知,在嵌入式应用领域,最大的特点是软硬件平台的多样性,以及应用对系统的功能和界面要求的多样性。移动终端中间件技术针对移动嵌入式系统,为应用开发人员提供了一个统一的开发平台,是为移动通信设备等嵌入式设备提供的运行平台,包含运行于终端设备的平台、模块、类库等。我们必须为它设计一个合理的体系结构,从而保证平台的稳定性和高性能,真正具有实用性。

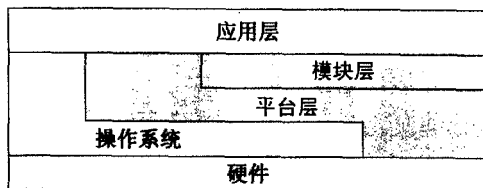


图 3.1 总体结构

上图中描述了基于移动终端中间件的终端设备的逻辑结构,图中阴影部分为移动终端中间件。移动终端中间件是建立在硬件和操作系统之上的,为应用层提供了开发的平台。我们将移动中间件划分为两个层面来理解:平台层和模块层。

3.3.2 平台层

平台层是移动中间件的底层部分,主要提供系统级的功能接口,同时为应用层和模块层提供接口支持。

平台层主要包括:文件系统管理、电话应用、通信协议接口、资源和电源管理、UI、安全、个人信息管理等功能接口。

3.3.3 模块层

模块层是移动中间件的上层部分,它将功能接口按照具体的业务进行封装,形成功能接口集,开发者可以针对某个应用选择相应的模块中的某个功能集进行开发。

模块层分为：基本功能集和增强功能集。基本功能集提供了构成移动终端的基本业务要求，可以作为必选项要求终端支持，这些业务大部分都已经在现有的终端上得到了一定规模的应用，而且取得了很好的市场效果；而增强功能集对软硬件的要求较高，可以作为可选要求终端支持。

随着通信技术和业务的发展，基本功能集和增强功能集的内容存在发生变化的可能，增强功能集中的某些 API 集可能会演进到基本功能集中，作为基本业务必选支持。某些业务也可能被淘汰，从功能集中删除（这依赖于该业务的市场情况而定）。

3.3.4 应用层

应用层是基于移动中间件平台（包括平台层和模块层）提供功能接口的直接面向用户的上层应用。应用层是移动业务的客户端软件的集合。

3.4 移动终端中间件的功能集

根据之前对移动业务的分析，我们将移动终端中间件按照业务功能划分为不同的功能集。功能集是一个完整的智能电话软件包，它能够提供强制基础部分的技术实现方法。使得应用软件和服务开发人员能够基于移动中间件开发智能电话产品，不仅能够为开发人员提供许多成熟的技术，而且使得设备厂商能够通过选择最适合于目标用户群的领先软件生产出差异化的产品。

本文主要介绍移动终端中间件为应用程序提供的功能接口，所以涉及到底层操作系统或硬件的诸如进程管理、中断处理、存储管理、设备管理、文件系统管理、网络协议及系统应用等，本文只做概要性的介绍，具体的实现将在移动终端中间件规范中定义。

3.4.1 平台功能集

(1) 进程管理

进程是可并发执行的程序在一个数据集合上的运行过程。进程管理是中间件的关键部分，它的设计和实现直接影响到整个系统的性能。在一个多进程的系统中，一个时间段内可以有多个进程“并发”执行。这样就避免了较为快速的 CPU 等待较为低速的 I/O 设备的情况，提高了 CPU 利用率，从而提高系统的性能。另一个方面，同时运行多个进程，就可以同时提供多种服务或者同时为更多的客户服务，这也是系统的基本任务。

(2) 中断处理

中断是指在计算机执行期间，系统内发生任何非寻常的或非预期的急需处理事件，使得 CPU 暂时中断当前正在执行的程序而转去执行相应的时间处理程序。待处理完毕后又返回原来被中断处继续执行或调度新的进程执行的过程。移动终端中间件的中断处理系统将中断正确路由到中断处理代码中的正确位置。中断发生时终端处理系统首先读取系统可编程中断控制器中中断状态寄存器判断出中断源，将其转换成中断数组中的偏移值。当调用设备驱动的中断处理过程时此过程必须找出中断产生的原因以及相应的解决办法。

(3) 文件管理

移动终端中间件文件管理系统支持多种文件系统，各种设备中都可以实现一个文件管理器，让用户创建、删除，及移动一些文件和目录。并且这些操作对于开发者到文件系统是透明的，即开发者无需了解存储设备是何种文件系统。文件管理功能集主要包括：文件信息 API，通用文件管理 API，通用文件对话框 API 等。

(4) 电话应用

语音业务依然是移动业务中的重要收入来源，电话应用是移动终端中间件的基本业务，同时，基于语音业务可以有很多诸如存储、同步等的增值功能。电话应用功能集主要包括：电话特性 API，电话服务和特性 API，快捷拨号 API，通讯记录 API，消息指示 API，情景模式 API，声音控制 API，声音识别 API 等。

(5) 用户界面

用户界面是直接面向用户的，用户界面的易用性、舒适度等直接影响到用户的业务体验。界面设计、软件界面设计、桌面设计、图标设计、按钮设计、菜单设计、导航条等等都需要中间件有一套强大的 UI API 来支撑。使得厂商能够自由地创建他们自己的定制用户界面。下面举例说明一些最常用的界面控件，以及他们的用途：

表 3.3 UI 控件列表

控件	说明
对话框	一个对话框是一个模型控制，它通常用来采集用户的数据。所示的对话框由一个标识和一个编辑器组成。一个对话框通常会有一个 OK 按钮和一个 Cancel 按钮。

查询	查询是一个简化的对话框, 包含一个或两个编辑器。查询提供了一条从用户获取数据的简便途径。它们有一个 OK 按钮和一个 Cance 按钮。
表格	表格能够提供以较复杂方式处理数据的能力。表格与对话框类似, 它可以包括其它用户界面元素, 例如: 编辑器, 弹出列表, 滚动条等。一个表格具备一个视图和一个编辑模式; 这对于需要存留的用户数据是有用的。例如, 一个联系簿应用软件可使用一个表格以便于用户查看联系簿详细信息并在必要的时候进行编辑。一个表格也可以由多页组成, 以使用来对相关控制进行分组。导航键(左/右)可用于页间切换。
弹出列表	弹出列表之所以如此命名, 是因为它们在屏幕的底部弹出。通常, 它们有 OK 和 Back 按钮。
列表框和网格	有多种可用列表类型: 单行、双行、带标题、编号和可标记型等。列表可以包含图形。
选项菜单	选项菜单是非常重要的用户界面组件。大多数多应用软件都需要创建一个选项菜单, 并对它产生的命令进行处理。

随着用户对于移动业务的体验要求越来越高, 终端的 UI 技术也必须不断的更新和发展, 以创造绚丽和实用的效果, 如: 动态桌面、个性化菜单、震动提示等。这就要求移动终端中间件的用户界面功能集更加强大和完善, 功能集主要包括: 通用 UI API, 震动控制 API, 触摸屏 API, UI 框架集, 动态 UI API, 扩展 UI API 等。

(6) 资源和电源管理

对于开发人员, 资源管理和代码效率极为重要。极其有限的内存意味着开发人员应尽可能地节省内存, 并遵守编码标准, 以把问题减到最少。移动终端中间件为资源和电源等的管理提供一系列支撑的功能。

栈是为每个应用软件预留的具有固定大小、遵循后进先出(Last-In, First-Out, LIFO)规则的系统内存区。该区域中存储临时数据, 如局部变量。在每个函数调用结束后, 该函数中生成的所有局部变量都随着栈的展开被销毁。就速度而言, 使用栈内存比访问堆要快。但是, 栈仅适用于少量数据的情况。堆是根据每个应用软件的需要, 由系统动态分配的内存区。堆内存可以在程序执行期间的任何点分配和释放。其生命周期由应用软件控制, 而不是由系统控制; 与栈内存中使用的 LIFO 系统不同, 堆内存可以以任意顺序释放。资源管理功能集主要包括: 资源管理 API, 资源释放 API, 内存管理 API, 电源管理 API。

不管是在栈中创建, 还是在堆中创建, 所有对象都具有相似的生命历程: 为对象分配内存、对象初始化、使用对象、释放对象使用的资源(清除)、释放内

存。适当的清除对象通常很重要，但是，对于移动设备，这尤其重要。在应用软件终止时，栈和堆被销毁，但是设备中的很多应用软件可能成周、成月或成年运行，因此保证没有内存泄漏至关重要。中间件有一个开发人员必须使用的清除支持 API，以确保在发生错误的情况下释放资源。由于在开发应用软件时内存管理非常重要，所以可以使用一些 API 方便地跟踪内存泄漏。电源管理 API 则可以监视电池电量；潜在的分配电量消耗，如：背景灯、通信、屏保、摄像机、计时器等。

(7) 通信

提供访问网络协议（如 TCP/IP）的 Sockets API，从而允许用户设计用来收发特定应用软件数据的程序。然而，这个 API 比大多数 API 的功能更多，它允许选择协议，包括 SMS 和红外线。

移动终端中间件提供对通信的广泛支持。这些 API 功能包括：使用 IrOBEX API 进行文件传送，使用 Telephony API 呼叫远程设备，使用串口或者 USB 等的 PC 间通信，使用红外的设备间传送文件，使用蓝牙的设备间传送文件，访问 Web 服务，通过 WAP 协议访问 WAP 服务。另外，需要具有本地连接的管理功能 API：连接管理 API，连接监视服务 API。

(8) DRM

由于移动业务涉及的图铃、游戏、程序往往都是由一定的内容提供商提供的，存在版权问题，因此数字版权管理对于业务的推广具有重要意义。

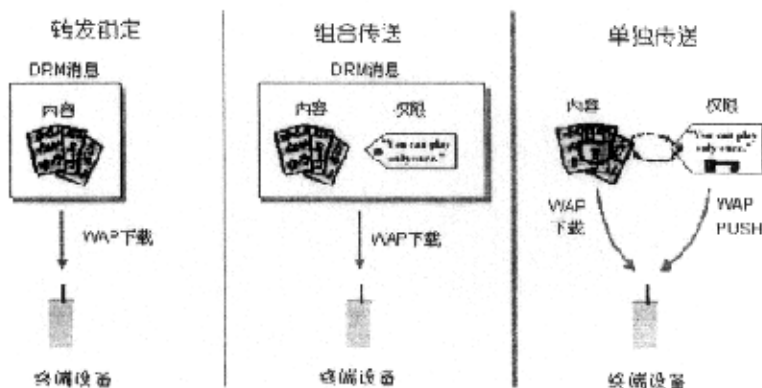


图 3.2 OMA DRM 1.0 的三种方式^[2]

DRM 功能集包括：DRM 通用 API，DRM 权限管理 API。根据 OMA DRM 协议 1.0 版本的规定，OMA DRM 允许根据内容提供商确定的内容使用权力，将

媒体内容封装到特定的 DRM 消息中。DRM 消息包含一个媒体对象和一个可选的使用权对象。限制用户对数字媒体对象的控制能力,以防止内容的非法拷贝。用户通过特定的客户端软件从 DRM 消息中恢复出媒体对象,并根据权限进行操作。OMA DRM 具有以下三个级别:

转发锁定:用户设备允许播放、显示或运行媒体对象、但不允许转发它们。对于转发锁定方法,DRM 消息中只包含媒体对象,用户没有播放、显示或运行以外的其他权限。

组合传送:允许在 DRM 转发锁定的基础上为媒体对象设置使用规则。它在 DRM 信息中添加了一个使用权对象。使用权定义了设备是如何运行内容的。可以采用时间和计数来限制使用权。此方法支持预览功能。

单独传送:用以保护高价值媒体并且支持超级发布功能,它允许设备转发媒体,但不包括使用权。这一方法采用单独信道传送媒体和使用权,因此比组合传送方式更安全。利用对称加密方法被将媒体加密成 DRM 内容格式(DCF),而使用权保存了内容加密密钥(CEK),这样,设备的 DRM 用户代理可以使用此密钥进行解密。

(9) PIM

所提供的 PIM 应用套件中包括日程安排(支持农历)、电话簿(Phonebook)、照相簿(Photo Album)、待办事项表(To-Do list),及各种文件管理应用等。这都是一些独立应用,都能方便地将与其它基于 PC 的 PIM 应用进行同步。PIM 模块的功能集主要包括:时钟 API,电话簿(Phonebook)API(包括 SIM 卡的 ADN),信息管理 API(包括 SIM 卡的 SMS),照相簿(Photo Album)API,日程安排 API,待办事项表(To-Do list)API,记事本 API 等。

(10) SyncML

SyncML 是一个基于 XML 的技术,它用于网络设备的通用数据同步。爱立信、IBM、Lotus、摩托罗拉、松下、诺基亚、Openwave、Starfish Software 以及 Symbian 都是 SyncML 标准的积极推动者,与支持成员一起促进了此技术的发展。SyncML 技术的目标是让任何移动设备的网络化数据都能实现同步并保证设备间同步的无缝连接。

SyncML 针对移动设备,移动设备可以不永远在线,但网络同步服务却永远在线。SyncML 还可以用于对等的的数据同步。在网络同步服务和移动设备以不同

的格式或使用不同的软件系统存储数据时, SyncML 特别有用。为了保证可交互性, SyncML 描述了通用数据格式是如何在网络上表示的。一旦需要, SyncML 允许增加新的数据格式,这保证了 SyncML 的可扩展性。这样不论移动设备是何种类型,运营商都能为他们的客户提供公共同步接口。SyncML 的第一个应用实现就是让拥有支持 SyncML 设备的用户通过同步始终拥有最新的日历和联系人信息。

移动终端中间件平台有一个 SyncML 客户代理和一个同步引擎,同步引擎基于 SyncML 规范并可用于无线空中(OTA)同步。OTA 同步通过 HTTP 和 WAP 协议栈工作:开始走 WSP 协议上,然后走 HTTP 协议。OTA 同步的特性有:客户发起端对联系人信息的双向同步(基于 vCard 对象类型);客户发起端对日历和待办事项列表信息的双向同步(基于 vCal 对象类型)。联系人数据格式遵循 vCard;日历的数据格式遵循 vCalendar。联系人名片能通过蓝牙、红外以及电子邮件交换。移动终端中间件平台从一开始就支持 OTA 同步(基于 WAP);后续版本也支持本地同步(通过基于蓝牙和红外的 Obex 协议)。同步功能在下阶段可能在备份、恢复以及服务器初始化同步方面有所加强。

带有 SyncML 服务器代理和同步引擎的 SyncML 服务器必须实现一个端对端系统。典型的解决方案由一个数据库服务器和一个应用服务器组成 SyncML 服务器。SyncML 服务器也要与其他网络元素整合,如运营商网络中的认证、计费以及配置模块。用户通常从终端发起同步会话。这时在 SyncML 客户端和 SyncML 服务器间建立了一个数据调用连接。当客户端通过认证后,数据交换开始。SyncML 服务器管理着同步进程,此间发生的步骤如下:新的数据加载到终端或应用服务器;从终端或应用服务器删除数据;修改的数据更新终端或应用服务器。未更改的数据不作交换,这样就节省了时间和宝贵的带宽。当操作完成后,服务器和客户端都要更新它们的日志文件以保证在下次同步会话前确保数据为最新。综上所述,我们将 SyncML 的 API 功能集定义为:通用同步装置 API,数据同步适配器 API,数据同步概要信息列表 API,数据同步装置 API。

3.4.2 模块功能集

(1) 基本功能集

a) SMS

该功能集支持与 3GPP (www.3gpp.org) 各标准兼容的 GSM/UMTS SMS。同时包括:能在一个单一的用户界面操作中发送和接收超长短消息(最长可达 450

个字符)。SMS 的 API 功能集如下: SMS I/O API, SMS Messaging API。

那么应用程序如何调用移动终端中间件的 API 发送 SM 呢? 我们来看一个例子: 发送一条文本消息到另外一个用户, 下面的代码将发送字符串“Hello World!”到另外一个用户。

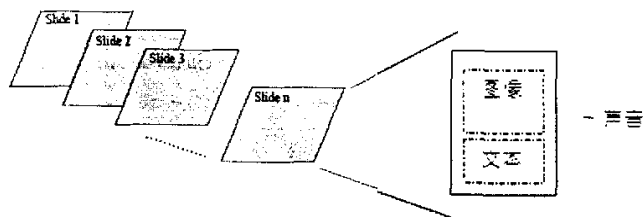
```
try {
    String addr = "sms://+358401234567";
    MessageConnection conn = (MessageConnection) Connector.open(addr);
    TextMessage msg =
        (TextMessage)conn.newMessage(MessageConnection.TEXT_MESSAGE);
    msg.setPayloadText("Hello World!");
    conn.send(msg);
} catch (Exception e) {
    ...
}
```

服务器响应并且接受消息: 下面的代码描述了服务器等待消息发送到 5432 端口并且响应他们。

```
try {
    String addr = "sms://:5432";
    MessageConnection conn = (MessageConnection) Connector.open(addr);
    Message msg = null;
    while (someExitCondition) {
        // wait for incoming messages
        msg = conn.receive();
        // received a message
        if (msg instanceof TextMessage) {
            TextMessage tmsg = (TextMessage)msg;
            String receivedText = tmsg.getPayloadText();
            // respond with the same text with "Received:"
            // inserted in the beginning
            tmsg.setPayloadText("Received:" + receivedText);
            // Note that the recipient address in the message is
            // already correct as we are reusing the same object
            conn.send(tmsg);
        } else {
            // Received message was not a text message, but e.g. binary
            ...
        }
    }
} catch (Exception e) {
    ...
}
```


b) MMS

多媒体消息业务 (MMS) 是短消息业务 (SMS) 和增强型消息业务 (EMS) 的进一步发展。多媒体消息不仅仅局限于文本信息, 还可以传递更为丰富的信息, 包括图像、音频和视频等。MMS 业务虽然在业务表现上类似于 SMS, 但在实现方法上采用 WAP 事件的处理流程。用户提交并在多媒体消息中心存储后, MMS 业务利用基于 SMS 的 WAP Push 下发通知, 接收方通过 WAP Pull 方式主动从多媒体消息中心下载多媒体消息。MMS 的 API 功能集如下: MMS 引擎 API, MMS 客户端 API, SMIL API 等。

图 3.3 多媒体消息的结构^[3]

MMS 引擎和客户端 API 分别实现了 MMS 的客户端协议栈以及 MMS 管理操作等功能。同步多媒体集成语言 SMIL (Synchronized multimedia Integration Language) 是基于 XML 的语言, 它可用于创建多媒体显示程序。SMIL 能够根据在屏幕上以及时间坐标轴上给出元素的位置, 将不同媒体元素, 例如图象、声频、视频和文本等合成到显示之中。多媒体消息服务 (MMS) 客户端能够支持所谓的 3GPP SMIL profile, 它是 SMIL 2.0 语言的一个子集。MMS 客户端支持各种媒体类型, 这些类型包括 GIF (包含动画) 和 JPEG 图象、MIDI 和 AMR 音频、以及 H.263 视频等。下面的代码是一个用 SMIL 语言描述的 MM 例子, 它定义了 MM 的内容和表现形式:

```
<smil>
  <head>
    <meta name="title" content="mms" />
    <meta name="author" content="John Smith" />
    <layout> <!--This an "landscape" screen (2*qcif)-->
      <root-layout width="352" height="144"/>
      <region id="Image" width="176" height="144" left="0" top="0" />
      <region id="Text" width="176" height="144" left="176" top="0"/>
    </layout>
    <!-- <layout> // This is a "portrait" screen -->
    <!-- <root-layout width="176" height="216"/> -->
    <!-- <region id="Image" width="176" height="144" left="0" top="0" /> -->
```

```

<!-- <region id="Text" width="176" height="72" left="0" top="144"/> -->
<!-- </layout> -->
</head>
<body>
  <par dur = "8000ms">
    <img src = "FirstImage.jpg" region="Image" />
    <text src = "FirstText.txt" region="Text" />
    <audio src = "FirstSound.amr"/>
  </par>
  <par dur = "7000ms" >
    <img src = "SecondImage.jpg" region="Image" />
    <text src = "SecondText.txt" region="Text" />
    <audio src = "SecondSound.amr"/>
  </par>
  <par dur = "4000ms" >
    <img src = "ThirdImage.jpg" region="Image"/>
    <text src = "ThirdText.txt" region="Text"/>
    <audio src = "ThirdSound.amr"/>
  </par>
</body>
</smil>

```

c) WAP

WAP 是无线应用协议（Wireless Application Protocol）的缩略形式，WAP 业务在 3G 业务中泛指 OMA WAP2.0 协议框架中所定义的移动数据业务。

WAP 业务使用类似于互联网的数据传输方式和业务实现模式，在无线环境下实现了各类互联网服务，如 WAP、Web 页面浏览，以及数据下载服务。同时通过 WAP 论坛定义的一系列增强功能，如 WAP Push、端到端安全、UA Prof，WAP 业务可以为用户提供丰富的移动互联网体验。OMA 定义的 WAP2.0 体系包含了大量的组件，下图显示的是 OMA WAP 规范中所要求的完整 WAP 协议栈架构。目前的浏览器应当是一个双模浏览器，它本身支持带有 WAP CSS 的 XHTML MP 和 WML 1.x。这意味着用 WML 1.x 编写的现有浏览应用软件以及新的 XHTML MP 服务均可在浏览器上运行。

带有 WAP CSS 的 XHTML MP 使得开发者在规划页面格式和布局方面有更多的控制权，因此，相对 WML 1.x 来说，应该优先考虑使用它。WML 1.x 能够用于一些特别的 WML 1.x 功能，例如：WML 事件或 WML 脚本调用等，如果有必要的话可以使用 WML 1.x。用 XHTML MP 和 WML 1.x 编写的页面可实现无缝连接。无论什么样的内容交叉融合在一起，它们的历史记录都将保存下来

以使用户能够返回到以前的页面。用户不需要知道他们目前所浏览的页面是由 WML 1.x 还是由 XHTML 编写的。

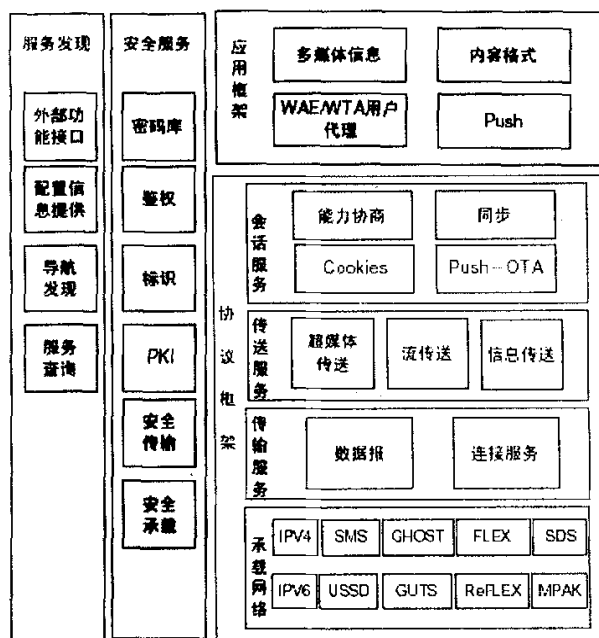


图 3.4 WAP 协议架构

转向使用 TCP/IP 协议栈是开发丰富的 2.5-3G 多媒体服务的主要因素之一。与 WAP 1.x 协议栈相比，TCP/IP 更安全并能够使得用户访问更广泛的 Internet 服务。所以，移动终端中间件的 WAP Browser API 功能集需要支持 WML、WML Script、XHTML 移动描述、TCP/IP、WAP 之上的 HTTP 1.1、WAP、WAP 推送、WAP 层叠样式表（Cascading Style Sheets, WAP CSS）、安全套接字层（Secure Sockets Layer, SSL）v3.0，以及传输安全层（Transport Layer Security, TLS）1.0。WAP 浏览器 API 功能集包括：Browser 引擎 API, Browser 工具和插件 API, HTTP 代理 API, Networking API, 接入点引擎 API, 接入点设置管理 API 等。

d) 多媒体

根据媒体类型的多样性制定多媒体功能集来规范终端设备的支持能力和业务表现形式。客户端功能单元可分为控制、场景描述，媒体编码以及媒体和控制数据的传送。

控制相关的单元包括会话建立，能力交换和会话控制，会话建立是指由浏览器或直接在终端用户界面输入 URL 地址发起 PSS 会话；能力交换可以根据不同的终端能力选择或适配媒体流。会话控制处理媒体流的建立，及开始，暂停，快

前、停止等功能。场景描述由空间结构及媒体演示中不同媒体间的时间关系描述组成。前者给出屏幕上不同媒体成份的布局，后者控制不同媒体的同步。媒体编码包括语音、音频、视频媒体的编码。

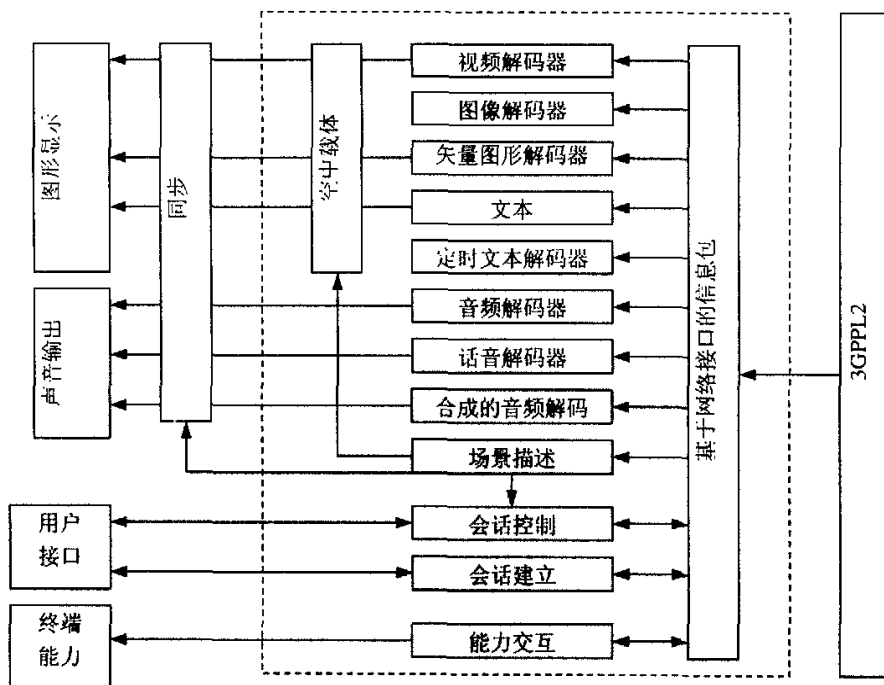


图 3.5 多媒体处理模型

多媒体功能集主要的一些应用，包括：Camera API、图像阅览 API、播放器 API、相册 API、录音机 API、3D API、SVG API、Flash API 等。一般来讲，多媒体的处理可以划分为两部分：处理数据传输协议和处理数据内容。在本功能集中，有两个高级对象：DataSource 和 Player，每个对象被封装为对应的多媒体处理部分：DataSource 对应协议处理部分；Player 对应数据处理部分。另外，在引入工厂设计模式后，Manager 从 DataSource 创建 Player，方便起见，Manager 也支持从 URL 或者输入流创建 Player，如下图所示：

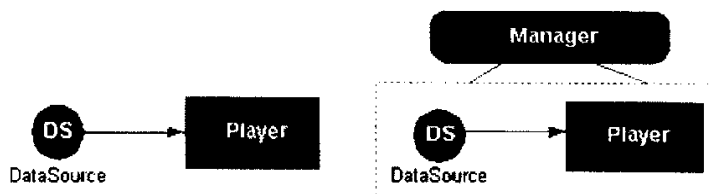


图 3.6 协议处理和内容处理^[4]

(2) 增强功能集

a) Bluetooth

蓝牙功能集从简单的聊天应用到激动人心的多人游戏，应用设计甚广，设备的同步多点连接特性使得各种一对多的应用成为可能。至于具体某一部移动终端中间件支持的手机是否具有蓝牙功能，则取决于该电话制造商的硬件实现——也就是说它是否包含有蓝牙模块。

移动终端中间件平台层自带有很多的应用软件接口，如电话本和日程表，第三方开发者的应用软件可以通过公开的 APIs 去调用这些应用的服务。中间件的多功能特性结合蓝牙的无线自由度可以为诸如多人游戏、短消息、聊天或者其他有趣的点对点应用等新型应用提供极其出色的平台环境。蓝牙还提供了开发各种无线附件应用的可能性，开发者可以为那些通过蓝牙使用的无线附件的智能电话开发应用，这些附件可能是条形码读取器、数字笔、健康监测设备、以及 GPS 接收器等。与其他许多通讯技术一样，蓝牙是由多个不同层次的组件构成，Bluetooth SIG 为其制定了蓝牙规范 v1.1，如下图所示。

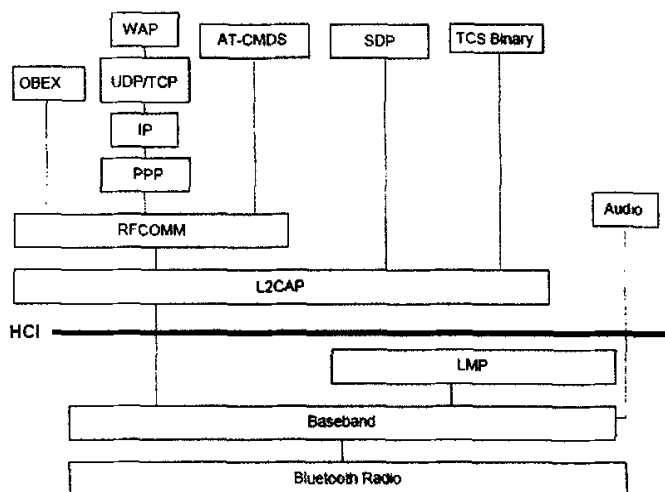


图 3.7 蓝牙的栈结构^[5]

蓝牙主机控制器模块 Bluetooth Host Controller 作为蓝牙的底层，是典型的硬件实现，应用程序（或在蓝牙上添加的服务）不能直接控制。蓝牙主机模块允许应用程序通过蓝牙连接发送/接受数据，或者配置蓝牙连接参数：

- RFCOMM 允许应用程序将蓝牙连接模拟为串口进行操作。这样可以支持传统的协议。
- 逻辑连接控制和适配协议 The Logical Link Control And Adaptation Protocol (L2CAP) 允许连接的 finer-grained 控制，它用来控制多用户的

连接，处理报文的分割和组合，这是应用程序的常用选择。

- 服务发现协议 The Service Discovery Protocol (SDP)用来定位和描述蓝牙设备提供或者可用的服务，应用程序主要在设置和另外的蓝牙设备通信时使用它。
- 主机控制接口 The Host Controller Interface (HCI)驱动包装了和硬件通信的高层模块。

概括以上分析，下面提到的三大类的 API 将为 Bluetooth 应用程序提供对 RFCOMM, L2CAP, SDP 和部分的 HCI 操作支持。

(1) 发现类 API

设备发现
服务发现
服务注册

(2) 设备管理类 API

通用访问概要信息
安全

(3) 通讯类 API COMMUNICATION

串口概要信息
逻辑连接控制和适配协议(L2CAP)
对象交换协议(OBEX)

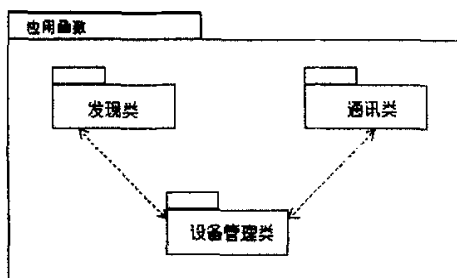


图 3.8 蓝牙各 API 模块间的功能关系

b) SIP

SIP (Session Initiation Protocol, RFC 3261) 是由 IETF (国际互联网标准制定组织) 制定的面向 Internet 会议和电话的信令协议。SIP 是一个应用层的协议，可以建立，修改或者中止多媒体会话或者呼叫。它是一个基于 ASCII 的端到端的协议，它实际上是在网络上提供“约会”服务。SIP 拥有明显的优越性：优异的可扩展性—大大提高了系统的处理能力；与 Internet 紧密结合—使通讯更加轻松便

捷的；卓越的开放性—不仅能够对手机、PDA 等移动设备提供良好的支持，对于在线即时交流、语音和视频数据传输等多媒体应用也能够很好地完成。

SIP 能够连接使用任何 IP 网络（有线 LAN 和 WLAN、公共 Internet 骨干网、移动 2.5G、3G 和 Wi-Fi）和任何 IP 设备（电话、PC、PDA、移动手持设备）的用户。基于 SIP 的应用（如 VOIP、多媒体会议、Push-To-Talk（PTT，按键通话）、定位服务、在线信息和 IM）即使单独使用，也会为服务提供商、ISV、网络设备供应商和开发商提供许多新的商机。不过，SIP 的根本价值在于它能够将这些功能组合起来，形成各种更大规模的无缝通信服务。

使用 SIP，服务提供商及其合作伙伴可以定制和提供基于 SIP 的组合服务，使用户可以在单个通信会话中使用会议、Web 控制、在线信息、IM 等服务。实际上，服务提供商可以创建一个满足多个最终用户需求的灵活应用程序组合，而不是安装和支持依赖于终端设备有限特定功能或类型的单分散的应用程序。无线一键通(PTT)在蜂窝网络中引入了直接点对点 and 点对多点语音通信服务。它使得用户能利用有魅力的移动设备享受普遍的双向无线服务，并由此促进移动服务，为实时语音通信领域注入新的商机。SIP 功能集包括：

- SipConnection
- SipClientConnection
- SipServerConnection
- SipConnectionNotifier
- SipClientConnectionListener
- SipServerConnectionListener
- SipDialog
- SipHeader
- SipAddress
- SipRefreshHelper
- SipRefreshListener
- SipException

应用程序使用 SIP API 执行 SIP User Agent (UA)功能，下图描述了终端 A 和 B 使用 SIP User Agent Client (UAC)和 User Agent Server (UAS)的工作原理^[6]。

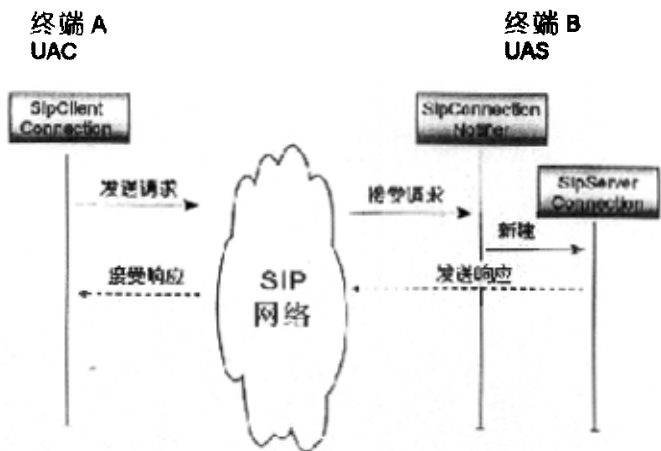


图 3.9 SIP 工作原理

实际上，在同一个终端上可以使用 SIP client 和 SIP server 连接，如下图中终端 A，这样可以执行 UAC 和 UAS 功能。

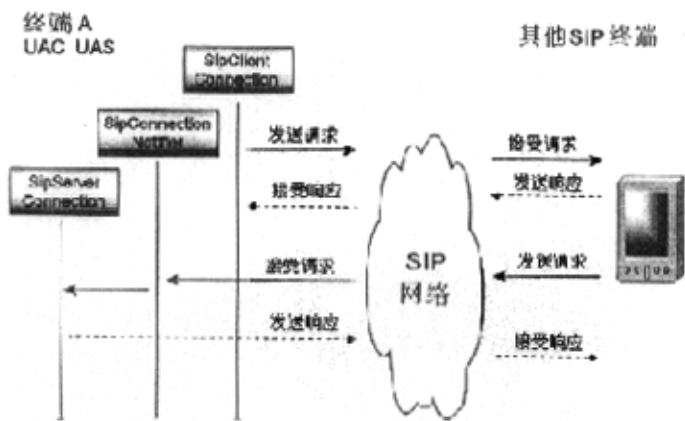


图 3.10 SIP 实际工作流程

c) LCS

位置业务（LCS）是通过移动网络获取移动终端用户的位置信息，内容供应商和业务供应商利用这个位置信息，结合 MMS、WAP 或 SMS 等为用户提供相应位置服务的一种增值业务。

移动终端中间件的 LCS 功能集是为终端基于移动位置应用而定义的，它的 API 设计用来产生当前地理位置信息。主要的功能包括：获得当前地理位置、终端定位、终端存储路标的访问等。API 功能集主要包括：

- AddressInfo
- Coordinates
- Criteria
- Landmark
- LandmarkException
- LandmarkStore
- Location
- LocationException
- LocationListener
- LocationProvider
- Orientation
- ProximityListener
- QualifiedCoordinates

LCS 功能集可以为基于上行链路的定位测量提供需要的信号, 或者通过下行链路信号的测量, 提供一些定位相关的测量数据。这些测量与所选择的定位方法相关。另外, UE 可以包含独立的定位功能, 如 GPS 接收器, 具有独立定位功能的 UE 可能会使用 RAN 的一些广播信息进行定位辅助。LCS 功能实体包括客户位置功能(LCF)、位置系统操作功能(LSOF)、位置信号测量功能(PSMF)及位置计算功能(PCF)。

客户位置功能(LCF)提供 LCS 客户和 LCS 服务器之间的逻辑接口。该功能对请求位置通知一个或更多的 UEs 是有关联的, 有规定"QoS" 和接受响应, 包含任一位置通知或失败指示。

位置系统操作功能(LSOF)对数据的提供, 位置能力, 数据相关的到客户和订阅(LCS 客户数据和 UE 数据)合法, 错误管理和履行管理 LCS。

位置信号测量功能(PSMF)负责上/下行链路信号测量, 测量结果用来计算 UE 的位置。这些测量可进行相关或辅助定位。UTRAN 里会有一个或多个 PSMF, PSMF 也可位于 UE、Node B 或独立的 LMU 中。

位置计算功能(PCF)负责 UE 位置的计算。这一功能基于收集到的信号测量结果, 选择某一算法来活动最终的位置估计。

d) Web Services

Web Services 是构筑支持移动终端网络数据服务的功能集，可用于开发能访问基于 XML 和 SOAP 的 Web 服务的移动终端。移动终端中间件的 Web Services 功能集主要包括两个部分：

1、支持 XML 的解析

XML 是应用发送结构化数据到移动终端的首选格式，为了避免在每个应用中都要编写 XML 的解析代码，故制定了 XML 的解析功能集。

2、基于 Web Services 的 SOAP/XML 访问

基于 Web Services，本功能集允许移动设备远程访问 XML，避免定义新的网络协议、格式，避免存在标准的重用。

e) 加密和信任服务

加密安全和信任服务功能集主要包括智能卡通信、数字签名服务和基本用户信任管理和通用加密库等功能。加密安全和信任服务功能集主要包括：

1、APDU API

支持使用 APDU 命令和智能卡中的应用进行通信；

2、JCRMI API

定义了 Java Card RMI 客户端 API，允许应用激活远程的 Java Card 对象；

3、PKI API

定义了应用层的数字签名和基本用户信任管理，PKI API 是独立于设备的，可以广泛复用。

4、CRYPTO API

定义了一系列加密算法 API，提供基本的加密操作，支持签名验证、解密、加密等。

3.5 版本控制和演进

移动终端中间件作为一个软件产品，随着移动业务的发展，不可避免的要进行修改和升级，那么对于软件版本演进的控制是必不可少的。

本论文规定的是移动终端中间件的功能规范，包含了满足现有业务以及业务

扩展需求的功能集，但不一定中间件的每一个版本都需要完全支持所有的功能集。中间件的实现可以分步骤逐个阶段完成：

首先，基于 JTWI 框架的运营商定制的 Java 终端；

无线领域的 Java 技术规范（Java Technology for the Wireless Industry 即 JTWI, JSR185）定义了移动终端设备 J2ME 的实现框架和技术要求。运营商根据自身业务需要，以 JTWI 为基础，可以扩展或者裁减，形成自己的 Java 终端标准。

第二，推出运营商制定的移动终端中间件；

经过上述的技术积累之后，加上移动运营商的运营经验，运营商可以推出移动终端中间件版本，同时设计参考实现和测试规范，由终端厂商或者终端软件厂商实现。

第三，升级移动终端中间件，推出新版本。

不断将新业务、新技术引入移动终端中间件，及时的推出新版本或者新的功能集，支持中间件良性发展。

3.6 平台无关性

移动终端中间件具有平台无关性和可扩展性，并被所有中间件设备共享。这意味着通常只需对软件进行一次开发，就可使它运行在具有相同 UI 的所有中间件设备上。若 UI 不同，为了把应用软件移植到新的平台并对其优化，需要额外的工作。一个典型的合理设计的应用软件应当包含 80% 的非 UI 代码和 20% 的 UI 平台特定代码。必须修改的是 UI 特定代码。一个合理设计的应用软件把所有 UI 代码与用于引擎、通信和网络的代码分离，并使它们相互独立。

基于移动终端中间件的开放平台将会导致巨大经济效益的产生。移动设备将不再仅限于与有限数量的其它设备（通常是同一个制造商的移动设备或 PC，后一种情况要通过连接软件）交互。用户将可以与使用移动终端中间件的任何设备交互。移动终端中间件的理念是把这一概念进一步提升，以实现运行在不同制造商生产的基于中间件的终端上的服务和应用软件（消息服务、浏览等）之间的完全兼容性。随着移动终端中间件的发展，运营商和所有获取中间件许可的公司约定在可能的地方保留二进制兼容性(BC)。这使得为平台的早期版本开发的应用软件能运行在使用后期版本的设备上。但是在将来，可能会不再保留 BC，这会影响到一些应用软件的可移植性。

移动终端中间件将逐步实现在 Symbian、WinCE、Linux、Palm、BREW 等各类终端平台的移植。应用对于硬件相关的移植工作量变得很小,开发人员将会因为能够进入更大市场而受益,也就是说,在更大的具有统一标准的应用软件市场中,多个制造商生产的电话使用相同的平台,开发人员可以自由地为这个单一的平台开发应用软件。因此,不同厂家将会开发出大量高质量软件,并且这些软件能以较低的价格下载到无线设备中。最终,价格的降低意味着竞争更加激烈,显然这将使消费者获益。其中,消费者的一种受益是有更多软件可供选择,这是由于移植费用的减小降低了软件开发人员进入市场的门限。另外,消费者还可以与更多的用户进行无缝交互。

3.7 安全策略和数字签名

所有应用程序分为非信任和信任这两类应用。非信任应用指设备不信任其来源和文件完整性的应用程序。所有的中间件实现必须支持非信任应用的运行。信任应用指设备信任其来源和文件完整性的应用程序,移动终端允许该应用访问设备的敏感性的 APIs 或受到保护的功能。

保护域(Protection Domain)定义了一个域以及一套许可(Permissions)和相关的用户交互模式。许可由一套 Allowed Permission 和一套 User Permission 组成。Allowed Permission 是指在应用套件与某一保护域联系起来的基础上,明确允许访问某一受保护 API 或功能的许可。User Permission 是指在应用套件与某一保护域联系起来的基础上,经提示用户并得到用户明确授权后才可访问某一受保护 API 或功能的许可。每一个 User Permission 的用户交互模式也是保护域策略定义的内容。

全局的安全策略(Security Policy)和保护域的安全策略称为设备安全策略。这是一个可由用户配置的基于安全保护域机制的安全规则(许可)用户可调的安全策略体系。在实际实现中,设备安全策略可以由运营商或制造商创建,采用静态形式,在手机出厂前预置。安全策略的物理存储和实现有关,任何实现必须要足以防止泄露和篡改。应用这个策略的安全认证过程是通过证书认证实现对域的认证,从而认证应用(应用已和签名的证书域绑定)。每一个保护域和相应的保护域根证书(Protection Domain Root Certificate)绑定。保护域根证书用于验证应用的证书属性的有效性,从而为信任应用匹配相应的保护域,即确定该应用(或域)的所有的许可授权,最终决定应用是否被允许运行。全部的保护域根证书预置在无线终端中,为避免用户更新证书的麻烦,终端的根证书的有效期为 10 年。

运营商为应用套件的文件签名，并将该证书和签名（base64 格式）插入文件的相应属性，最终将应用套件提供给用户下载。用户从本地或者 OTA 下载服务器发现应用，将其下载到手机中。AMS 根据文件中是否出现对应的属性，来判断是否需要执行认证。如果没有该属性，则应用程序作为非信任应用套件进入非信任域并被安装。系统应用非信任策略对非信任域进行运行时访问控制。如果文件中出现了对应的属性，AMS 将执行认证。首先用预置在手机中的保护域的根证书来验证文件中由某个属性标识的证书的有效性。如果证书有效，则将该应用和相应的保护域绑定起来，然后从文件属性中取出签名（base64 格式）并解码。用该证书中的公钥、文件签名解密并验证签名。如果签名验证成功，表明该文件来源可信而且完整，可以进行安装。进行下载的安装过程。

接下来 AMS 对通过证书认证的信任应用套件进行授权：判断该应用套件要求的许可（文件中的权限属性的列表）和对应属性是否相同。如果不同，该应用套件不能被安装或调用；将该应用套件要求的许可和设备上的 Permissions 作比较（设备上的许可是指设备上所有受保护的 API 或功能的许可的集合）；将该应用套件要求的许可和该应用套件被绑定的保护域的许可信息（该保护域的 security policy，由一套 Allowed 和 User Permissions 组成）作比较，从而确定该应用套件能够使用的 APIs。成功的授权结果为应用套件得到它所要求的受保护的 APIs 和功能，否则终端提示用户相关警告或出错信息。下图描述了签名和认证的流程：

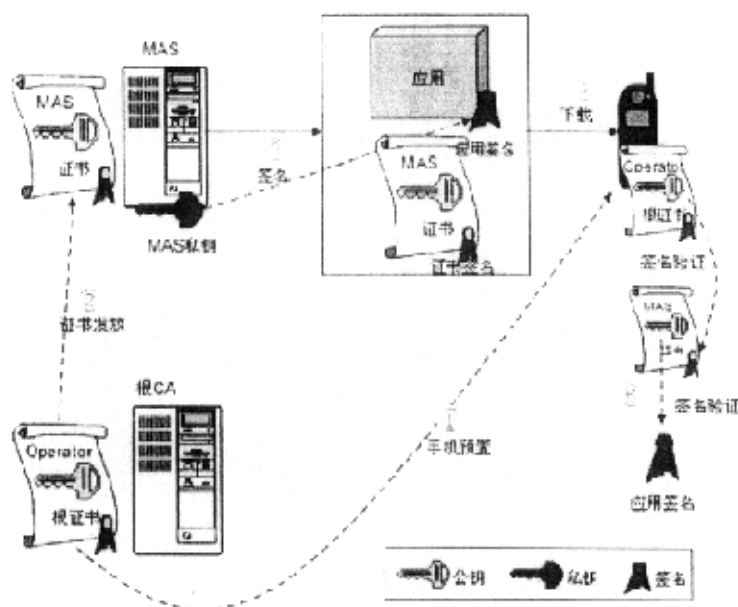


图 3.11 证书签名与认证流程

3.8 应用管理软件

AMS (Application Management Software) 是运行在终端设备上的应用管理软件, 其主要功能是:

- 1、从网络或其它环境下载应用;
- 2、在设备上安装、更新和删除应用;
- 3、启动、停止、悬挂和恢复应用。

一个典型的终端应用的生命周期如下图所示:

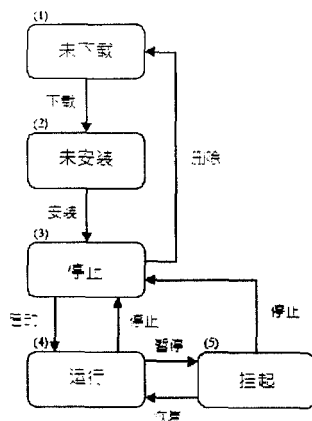


图 3.12 应用程序生命周期

3.9 应用开发

应用开发主要由终端厂商、终端软件厂商、CP/SP 等完成, 运营商需要提供针对移动中间件平台的应用开发工具 SDK 以及模拟器等构成应用开发环境。可以将中间件的 SDK 集成到现有的 IDE, SDK 允许扩展支持新的功能模块。逐渐形成以运营商为核心的开发社区。

对于不同的开发者, 运营商授予移动中间件不同的开发和使用权限, 分别设置平台层和模块层功能集的权限, 只有取得运营商或者用户的许可, 某些设计到操作系统、中间件安全等的敏感函数才可以使用。开发者完成应用开发后, 需交由运营商审查和许可授权发布。对于部分不涉及安全、个人隐私或者资费的应用, 运营商可以下放发布权利。

3.10 中间件的测试

测试包括移动终端中间件测试和应用测试两大部分。运营商需要成立一个终端中间件的测试中心，或者由开发联盟成立一个独立的测试中心，用来测试中间件以及应用程序的功能、性能等技术指标。

中间件测试包括平台层和模块层两部分的测试。测试需要针对每个功能集制定相应的测试规范，满足某个功能集的测试需要通过该功能集测试。即，中间件测试需要满足两个层面的要求：

- 规范一致性测试：主要是测试中间件平台的功能集是否符合规范，即中间件的正确性和准确性。
- 性能测试：运营商发布性能测试工具，也可以在社区公布测试排名，利用一系列的性能指标衡量中间件。

应用测试也包括以上两个层面的测试，另外，运营商需要对应用程序进行资格测试，审查应用程序是否含有非法、或者不健康的内容等，维护健康的生态链。

3.11 小结

本章详细介绍了以运营商为主导的移动终端中间件技术。分析了移动业务的需求，得出移动终端中间件的目标，提出了移动终端中间件的总体架构和技术方案。本章重点分析了移动终端中间件的功能集合，根据业务定义了较为完整的功能集划分、类库以及 API 等，并给出了部分功能集的实现原理。另外，本章还就中间件的版本演进、移植、安全、应用管理、应用开发以及测试做了比较详细的介绍。

第四章 基于 Java 的移动终端中间件的实现

4.1 软件环境的实现

将 Java 技术应用于嵌入式系统,则体系结构有所改变,这也是本课题所涉及的。在 Java 嵌入技术中,主要编程语言是 Java,采用类似标准 Java 的系统结构,由 Java 核心结合嵌入 Java API(扩展类库)组成新的 Java 程序运行环境,运行在 Java 虚拟机上,可用于各种不同的操作系统和处理器。

4.1.1 J2ME 软件架构

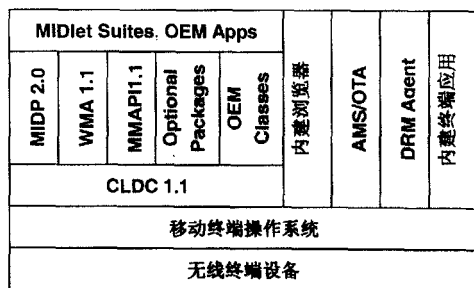


图 4.1 Java 移动终端软件架构

CLDC1.1

Connected Limited Device Configuration(CLDC)由一个虚拟机和一个最小类库组成,为一个特定范围的设备提供基本功能。
Connected Limited Device Configuration 1.0 (JSR-030)

MIDP2.0

Mobile Information Device Profile 2.0 (JSR-118)

WMA1.1

Wireless Messaging API 1.1 (JSR-120)

MMAPI1.1

Mobile Media API 1.1 (JSR-135)

Optional Packages

由运营商对 JTWI 进行采纳和扩展的 Java 功能包。

OEM Classes

由终端设备制造商自行采纳或扩展的 Java 功能包。

内建浏览器

指基于 WAP 协议栈的 WAP/WML 浏览器,或者是指基于 HTTP、TCP/IP 的超文本 (HTML/XHTML 等) 浏览器。

AMS/OTA

支持 Java 应用程序的下载、安装、执行、更新、删除、挂起和恢复,以及运行环境 and 应用、业务等的参数设置,管理和 DRM Agent 的交互等。

DRM Agent

数字版权管理代理参见[OMA DRM 1.0]。

(1) CLDC 1.1

终端设备支持 CLDC 1.1, 此外, 终端还必须支持:

最小应用线程数

对于某一 MIDlet suite, 允许其至少创建 10 个并发线程。

最小时钟分辨率

对于全部的 `java.lang.System.currentTimeMillis()` 调用, 所返回的时间和调用时刻之间误差值其中的 80% 不大于 40 毫秒。

时区标识习惯

语法:

```
CustomID:  
GMT Sign Hours : Minutes  
GMT Sign Hours Minutes  
GMT Sign Hours  
Sign: one of:  
+ -  
Hours:  
Digit  
Digit Digit  
Minutes:  
Digit Digit  
Digit: one of:  
0 1 2 3 4 5 6 7 8 9
```

小时必须在 0 到 23 之间, 分钟必须在 00 到 59 之间。例如 “GMT+10” 和 “GMT+0010” 分别意味着超前 GMT10 小时和 10 分钟, 格式是不依赖地区 (Locale) 的。数字必须来自 Unicode 标准的 Basic Latin 块。不能表示出夏时制。创建 `TimeZone` 对象时, 按下面方法格式化标识:

NormalizedCustomID:
GMT Sign TwoDigitHours : Minutes
Sign: one of
+ -
TwoDigitHours:
Digit Digit
Minutes:
Digit Digit
Digit: one of
0 1 2 3 4 5 6 7 8 9

例如,TimeZone.getTimeZone("GMT-8").getID() 返回
GMT-08:00.

编码名

使用 IANA (<http://www.iana.org/assignments/character-sets>).

字符的属性

支持字符属性(排序、大小写等)和 Unicode3.0 的“Basic Latin”
and “Latin-1 Supplement”区块。必须支持 Unicode3.0 对应中文
GBK 的区块。

Unicode 版本

支持 Unicode 字符集, 基于 Unicode3.0 标准。具体操作上必须支持
ISO Latin-1 字符和 GBK 字符。

(2) MIDP 2.0

终端设备支持 MIDP2.0, 此外, 终端还必须支持:

最少 Record Store 数

一个 MIDlet suite 可以创建至少 5 个互相独立的 Record Store。

媒体内容的 HTTP 支持

符合 MIDP2.0, 同时 HTTP 实现必须对所有支持的媒体类型
(Content-type) 提供支持。

定时器(Timer)分辨率

同 CLDC1.1 中的“最小时钟分辨率”。

最少定时器(Timer)数

每一 MIDlet 套件可以创建至少 5 个定时器, 不受“最小应用线程
数”的影响。

位图最小值

必须支持色深为 1、2、4、8、16、24 以及 32 比特像素的 PNG 图

片的装载,对于 JPEG 图片格式,同样等级的色深,终端设备必须支持等于 32768 总象素的适应性实现。

TextField、TextBox、Phone Book 的结合

当用户编辑一个文本框和文本域,并且文本框和文本域的限制是 TextField.PHONENUMBER,终端应支持从终端电话本中选择电话号码的机制,当电话本不可访问时,可不支持该机制。

TextField、TextBox 支持的字符

参见 [JTW1.0] 中的“Supported Characters in TextField and TextBox”。

邮件地址、URL 中支持的字符

有 TextField.EMAILADDR 和 TextField.URL 任何一个限制的类的实例,在字符输入方面,应与 TextField.ANY 一样。E-mail 地址和 URL 可以包含任何字符,但允许应用任意限制哪些特定的字符可以用在 URL 和 E-mail 地址中。

Push Registry Alarm 事件

MIDlet 应用应实现基于告警的 push 注册登陆。如果不存在其他的安全机制,若没有明确的用户许可,不允许使用 PushRegistry 告警功能。

(3) WMA 1.1

终端设备支持 WMA1.1,此外,终端还必须支持:

移动终端 SMS 的支持

终端通过使用在 JSR-120 规范附录 A 中定义的 API,支持短消息业务。

小区广播的支持

如果应用通过 Java API 接入小区广播,则应遵循 WMA1.1 附录 B 中的规定。

SMS Push

终端应按照 WMA1.1 附录 D 中的定义,为 SMS 协议支持 MIDP2.0 Push 处理。如果不存在其他的安全机制,若没有明确的用户许可,不能允许 PushRegistry SMS Push 功能。

(4) MMAPI 1.1

终端设备支持 MMAPI1.1,此外,终端还必须支持:

HTTP1.1 协议

为下载媒体文件（文件中包含终端支持的媒体格式），终端应支持 HTTP1.1 协议。

MIDI 特征集及其控制

应实现 MMAPI 中指定的 MIDI 特征集；应支持 MIDI 文件重放、音量控制。

视频快照的 JPEG 编码

支持视频特征集和视频图像捕获的应用应在视频快照中支持 JPEG 编码。

音信号序列（Tone Sequence）文件格式

应支持音信号序列文件格式。

(5) 其它

可选支持其它的功能集，如：

PDA Optional Packages for the J2ME Platform（JSR 75）

Java APIs for Bluetooth（JSR 82）

Location API for J2ME（JSR 179）

Mobile 3D Graphics API for J2ME（JSR 184）

4.1.2 应用管理软件

(1) 应用分析

支持对 JAD、JAR、MF 文件的分析处理。

Java 应用程序的 JAD 信息包括：

应用程序名称（MIDlet-Name）

应用程序版本号（MIDlet-Version）

开发商名称（MIDlet-Vendor）

AR 文件的 URL（MIDlet-Jar-URL）

JAR 文件的大小（MIDlet-Jar-Size）

适用终端类型（MicroEdition-Configuration/MicroEdition-Profile）

应用程序简介（MIDlet-Description）.....

(2) KVM 控制

支持 KVM 挂起、恢复等功能。如果有外部中断发生，那么可以将 KVM 挂起，KVM 的挂起应该不影响 Java 应用程序，当 KVM 恢复之后，Java 应用程序

可以继续执行。

(3) 中断处理

当应用管理器或 Java 应用处于激活和操作状态中时,如果有外部事件需要控制用户接口(如有语音呼叫时),或者有其它事件要终端结束或暂停当前的应用,应对用户进行语音或文本提示。

在发生事件要结束或暂停当前应用时,应提示用户选择接受或忽略该事件。在用户接受该事件时应保留现场并挂起应用或 KVM。在用户接受该事件并处理完成该事件后,应提示用户是否返回现场。这些事件包括:

- 来话呼叫;
- 接收到 SMS、MMS 等;
- 设备状态告警(如低电量提示);
- 预置提示(如闹钟)
- 其它各种外部中断或本地中断。

(4) 状态报告

在对 MIDlets 进行下载、安装、更新、删除等操作时,终端设备要发送状态报告。状态代码和状态信息见接口要求。

4.2 开发实例解析

下面,我们来看一个最简单的 MIDlet 程序实例—HelloWorld.java。程序代码清单如下:

```
package examples;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class HelloWorld extends MIDlet implements CommandListener
{
    private Command exitCommand;
    private TextBox tb;

    public HelloWorld(){
        exitCommand = new Command("Exit", Command.EXIT, 1);
        tb = new TextBox("Hello MIDlet", "Hello, World!", 15, 0);
        tb.addCommand(exitCommand);
        tb.setCommandListener(this);
    }
}
```

```

protected void startApp(){
    Display.getDisplay(this).setCurrent(tb);
}

protected void pauseApp(){

}

protected void destroyApp(boolean arg0){

}

public void commandAction(Command cmd, Displayable display){
    if (cmd == exitCommand){
        destroyApp(false);
        notifyDestroyed();
    }
}
}

```

程序说明:

HelloWorld.java 文件以下面的代码开始, 这些代码导入稍后会在 HelloWorld 类中使用的类:

```

package examples;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

```

由于 HelloWorld 类是一个 MIDP 应用程序, 它扩展了 MIDlet。并且实现 CommandListener 接口来处理按键事件:

```

public class HelloWorld extends MIDlet implements CommandListener

```

下面的方法是一个缺省构造函数, 用于添加 EXIT 按钮和 HelloWorld 文本框:

```

public HelloWorld(){
    exitCommand = new Command("Exit", Command.EXIT, 1);
    tb = new TextBox("Hello MIDlet", "Hello, World!", 15, 0);
    tb.addCommand(exitCommand);
    tb.setCommandListener(this);
}

```

通过调用 startApp() 方法启动应用程序, 在 MIDlet 的一次执行中它可能会被系统多次调用。如果 MIDlet 暂停, pauseApp() 将会被调用。要重新启动 MIDlet,

需要用 startApp()。所以只需执行一次的初始化代码应该放置在构造程序中：

```
protected void startApp(){
    //获得显示的引用并且显示HelloWorld文本框
    Display.getDisplay(this).setCurrent(tb);
}
```

pauseApp()被调用使得 MIDlet 处于暂停状态。destroyAPP()被调用，使得 MIDlet 处于销毁状态。在此应用程序中，当进入暂停状态或者销毁状态时，我们没有执行任何操作；但是我们仍然需要在 MIDlet 中实现 pauseApp()和 destroyApp()方法，因为他们是父 MIDlet 类中的抽象方法。

```
protected void pauseApp(){
}
protected void destroyApp(boolean arg0){
}
```

commandAction()方法是事件处理程序，被请求实现 CommandListener 接口。目前，它只接受电话的 Exit 按键输入，使程序进入销毁状态并通知 Java 应用程序管理器本 MIDlet 程序已经结束。

```
public void commandAction(Command cmd, Displayable display){
    if (cmd == exitCommand){
        //销毁这个MIDlet应用程序
        destroyApp(false);
        //通知Java应用程序管理器本MIDlet程序进入销毁状态
        notifyDestroyed();
    }
}
```

当在中国电信 MIDP 设备模拟器上运行时，这个最简单的 MIDlet 的输出如下图所示：



图 4.2 HelloWorld.java 运行结果

在这个图中，MIDlet 的 startApp()方法刚刚执行完成。如果按下 Exit 命令就会运行 commandAction()方法以及运行 destroyApp()和 notifyDestroyed()方法，结束这个 MIDlet。

这个例子虽然非常简单，但是说明了 MIDlet 程序的几个关键点：

- 1) 将这个 MIDlet 作为一个单独的包：package examples;
- 2) 通过 public class HelloWorld extends MIDlet implements CommandListener{}定义了这个 MIDlet 的主类 HelloWorld，并扩展了 Javax.microedition.midlet.MIDlet 类。由于 MIDlet 类处于 Javax.microedition.midlet 中，所以第一步就引入 Javax.microedition.midlet 包。
- 3) 为这个 MIDlet 子类生成了恶意个构造程序，public HelloWorld(){}, 利用它完成了应用程序的初始化工作。
- 4) 通过定义 startApp()、pauseApp()、destroyApp()三个方法实现了与底层的 Java 应用管理器之间的接口。这三个方法分别被 Java 应用管理器用来向该 MIDlet 程序发出信号，通知它进入某一状态。
- 5) 实现了特定的功能：在屏幕上显示了文本框 Hello,World，并接收用户按键结束程序。

任何一个标准的 MIDlet 应用程序都包括这五部分，以此构成一个完整的 MIDlet 应用程序。为了让一个 MIDlet 程序实现某种具体的功能，比如手机上的股票交易、机票预定、游戏等等，还必须通过使用 J2ME 提供的高级/低级用户界面库、网络接口库、永久存储库等各种方法才能得以实现。

4.3 小结

本章介绍了基于 Java 的移动终端中间件的实现。Java 技术在嵌入式系统的发展有相当大的优势，目前，已经支持了很多的移动业务。JTWI 定义了 Java 移动终端的系统结构和实现方式，运营商根据 JTWI 可以修改并扩展 API，这正是移动终端中间件的初级阶段。另外，本文以一个实例介绍了应用程序的开发过程，提出了基于移动终端中间件的开发模式。

第五章 总结和展望

5.1 总结

本文从理论和实践两个方面对移动终端中间件进行了深入的研究,提出了运营商主导的移动终端中间件的技术方案,并论证了基于 Java 的中间件实现。

第一章介绍了终端总体结构和技术、移动业务以及移动终端中间件的研究背景、意义和本论文的主要贡献。

第二章介绍了现有中间件技术,分析了移动终端中间件技术和传统中间件技术的联系和区别,总结了移动终端中间件技术的特点和所涉及的关键技术。

第三章详细介绍了以运营商为主导的移动终端中间件技术。提出了中间件技术方案的总体框架、中间件的功能集定义等,并论述了部分功能集的实现原理,另外,就中间件在发展过程中需要注意的各种问题进行了分析。

第四章介绍了基于 Java 的移动终端中间件的实现。运营商根据 JTWI 可以修改并扩展 API,逐步推出自己的移动终端中间件,另外,以一个实例介绍了基于移动终端中间件的开发方法。

5.2 展望及未来工作

随着无线数据应用平台的日益广泛的应用和应用复杂度的增加,移动终端中间件技术会得到更加广泛的应用。本课题对移动终端中间件的研究工作只是为今后奠定了基础,以后还需要在以下几方面作进一步的研究:

- 1) 制定和完善移动终端中间件功能集规范,发布功能规范;
- 2) 制定移动终端中间件技术实现规范,根据功能规范设计中间件的实现;
- 3) 联合厂商开发移动终端中间件参考设计和一致性测试规范;
- 4) 移动终端中间件的集成、推出试用版本;
- 5) 建立开发社区,设计业务组合和开发应用;
- 6) 移动终端中间件的商用、推出相应服务。

附 录

在校期间主要科研项目：

1. 中国电信 3G 终端业务研究（参与制定 Java 业务规范等）
2. 中国固网和 PHS 终端联盟小灵通机卡分离（参与制定 PIM-PS 接口规范等）

在校期间发表论文：

1. 刘向辉，黄国兴：《基于 PC/SC 的智能卡操作在 VC++.net 中的实现》，微型计算机应用，2005.4

参考文献

- [1] 3GPP:《TS23.101. General UMTS Architecture》, 2004
- [2] OMA:《OMA Digital Rights Management》, 2004
- [3] OMA:《OMA Multimedia Messaging Service》, 2004
- [4] JCP:《JSR 118. Mobile Information Device Profile》, 2002
- [5] JCP:《JSR 82. Java APIs for Bluetooth1.0》, 2002
- [6] JCP:《JSR 180. SIP API for J2ME》, 2003
- [7] Michael Barr:《C/C++ 嵌入式系统编程》, 中国电力出版社, 2001
- [8] Frank VahidTony Givargis:《嵌入式系统设计》, 北京航空航天大学出版社, 2004
- [9] 李华颺等:《精通 Java 中间件编程》, 中国水利水电出版社, 2003
- [10] 马忠梅:《ARM & Linux 嵌入式系统教程》, 北京航空航天大学出版社, 2004
- [11] Bill Venners:《深入 Java 虚拟机》, 机械工业出版社, 2003
- [12] 卜佳俊:《深入 BREW 手机游戏开发》, 清华大学出版社, 2004
- [13] 黄聪明:《Java 移动通信程序设计》, 清华大学出版社, 2002
- [14] 李迅:《J2ME 无线设备编程》, 机械工业出版社, 2002
- [15] Robert Lov:《Linux 内核完全注释》, 机械工业出版社, 2004
- [16] Bruce E.Krell:《POCKET PC 开发指南》, 清华大学出版社, 2003
- [17] Bruce Eckel:《Java 编程思想(英文版·第3版)》, 机械工业出版社, 2004
- [18] James Keogh:《J2ME 开发大全》, 清华大学出版社, 2004
- [19] 许江伟,陈平:《BREW 平台上的手机软件开发》, 计算机应用, 2003
- [20] 孙开翠,王汝传,杨立扬:《J2ME 中 CLDC 的安全性机制的研究》, 通信技术, 2003
- [21] 柯科峰,邵世煌,张珏:《基于 J2ME 的小型移动商务系统的设计与实现》, 计算机应用研究, 2004
- [22] 王东民,应忍冬,徐国治:《基于 J2ME 技术的嵌入式系统的应用开发》, 计算机工程, 2003
- [23] 江颖,赖梅,蔡家楣:《基于 J2ME 技术的移动证券研究与实现》, 计算机工程与应用, 2004
- [24] 王晋桃,朱欣焰:《基于 Java 手机的地理信息服务探索》, 测绘通报, 2004
- [25] 刘翔宇,徐进:《企业应用中移动技术的研究和实现》, 微计算机应用, 2004
- [26] 樊自甫等:《未来的移动互联网技术》, 通讯世界, 2003
- [27] 江颖等:《无线安全中间件的设计与应用》, 计算机系统应用, 2004
- [28] 郭忠文,汪治,冯业伟:《基于移动代理中间件的分布式计算研究》, 计算机应用研究, 2003

- [29] OMA: 《OMA Browsing》, 2004
- [30] 3GPP: 《TS 22.140. Multimedia Messaging Service》, 2004
- [31] 3GPP: 《TS 32.235. Charging data description for application services》, 2004
- [32] JCP: 《JSR 185. Java™ Technology for the Wireless Industry》, 2003
- [33] JCP: 《JSR 139. Connected Limited Device Configuration 1.1》, 2003
- [34] JCP: 《JSR 120. Wireless Messaging API 1.1》, 2003
- [35] JCP: 《JSR 135. Mobile Media API 2.0》, 2003
- [36] JCP: 《JSR 177. Security and Trusted API for J2ME》, 2004
- [37] JCP: 《JSR 179. Location API for J2ME》, 2003
- [38] JCP: 《JSR 172. J2ME Web Service Specification》, 2003

后 记

这篇论文是在我的导师黄国兴教授的悉心指导下完成的，在论文完成之际，首先向黄国兴教授致以衷心的感谢和崇高的敬意！在就读研究生期间，黄国兴教授给了我无微不至的教导和关怀。他对学科前沿的敏锐把握和对新技术的前景判断能力深深地影响了我，对学术问题的严谨认真、精益求精的态度使我获益匪浅。所有这些都将使我终生受益，再次对黄国兴教授表示深深的谢意！

同时，我要衷心谢谢上海电信技术研究院的林正汉。在研究院实习期间，林正汉作为我的企业导师，不但在工作和学习上给予我很多帮助和指导，而且他在生活中也对我关怀备至。林正汉对科研的严谨认真，以及对实习学生的关心体贴都给我留下了深刻的印象。感谢在实习期间所有曾经给予我帮助和教诲的上海电信技术研究院的各位领导和同事。

衷心谢谢同在研究院实习的各位同学，论文的完成离不开他们的帮助。还有和我朝夕相处的同学们，是他们帮助我顺利过完了研究生生活。

这篇论文也要献给我最重要的人——我的父母和家人。是他们给我创造了良好的生活环境和学习环境，也教给我很多做人的道理。在我成长的二十多年中，是他们的鼓励和帮助使我获得不断进取的动力！

基于Java的移动终端中间件的研究与设计

作者：[刘向辉](#)
学位授予单位：[华东师范大学](#)
被引用次数：11次

参考文献(38条)

1. [3GPP TS23.101.General UMTS Architecture](#) 2004
2. [OMA OMA Digital Rights Management](#) 2004
3. [OMA OMA Multimedia Messaging Service](#) 2004
4. [JCP JSR 118.Mobile Information Device Profile](#) 2002
5. [JCP JSR 82.Java APIs for Bluetooth1.0](#) 2002
6. [JCP JSR 180.SIP API for J2ME](#) 2003
7. [Michael Barr C/C++嵌入式系统编程](#) 2001
8. [Frank Vahid.Tony Givargis 嵌入式系统设计](#) 2004
9. [李华鹰 精通Java中间件编程](#) 2003
10. [马忠梅 ARM & Linux嵌入式系统教程](#) 2004
11. [Bill Venners.曹晓钢.蒋靖 深入Java虚拟机](#) 2003
12. [卜佳俊 深入BREW手机游戏开发](#) 2004
13. [黄聪明 Java移动通信程序设计](#) 2002
14. [李迅 J2ME无线设备编程](#) 2002
15. [Robert Lov Linux内核完全注释](#) 2004
16. [Bruce E Krell.俞忠东.石柱 POCKET PC开发指南](#) 2003
17. [Bruce Eckel Java编程思想](#) 2004
18. [James Keogh J2ME开发大全](#) 2004
19. [许江伟,陈平 BREW平台上的手机软件开发](#)[期刊论文]-[计算机应用](#) 2003(10)
20. [孙开翠,王汝传,杨立扬 1J2ME中CLDC的安全性机制的研究](#)[期刊论文]-[通信技术](#) 2003(8)
21. [柯科峰,邵世煌,张钰 基于J2ME的小型移动商务系统的设计与实现](#)[期刊论文]-[计算机应用研究](#) 2004(2)
22. [王东民,应忍冬,徐国治 基于J2ME技术的嵌入式系统的应用开发](#)[期刊论文]-[计算机工程](#) 2003(9)
23. [江颜,赖梅,蔡家楣 基于J2ME技术的移动证券研究与实现](#)[期刊论文]-[计算机工程与应用](#) 2004(6)
24. [王晋桃,朱欣焰 基于Java手机的地理信息服务探索](#)[期刊论文]-[测绘通报](#) 2004(3)
25. [刘翔宇,徐进 企业应用中移动技术的研究和实现](#)[期刊论文]-[微计算机应用](#) 2004(2)
26. [樊自甫,张红,杨俊蓉,万晓榆 未来的移动互联网技术](#)[期刊论文]-[通讯世界](#) 2003(7)
27. [江颜,蒋融融,蔡家楣 无线安全中间件的设计与应用](#)[期刊论文]-[计算机系统应用](#) 2004(4)
28. [郭忠文,汪治,冯业伟 基于移动代理中间件的分布式计算研究](#)[期刊论文]-[计算机应用研究](#) 2003(9)
29. [OMA OMA Browsing](#) 2004
30. [3GPP TS 22.140.Multimedia Messaging Service](#) 2004
31. [3GPP TS 32.235.Charging data description for application services](#) 2004
32. [JCP JSR 185.JavaTM Technology for the Wireless Industry](#) 2003
33. [JCP JSR 139.Connected Limited Device Configuration1.1](#) 2003
34. [JCP JSR 120.Wireless Messaging API 1.1](#) 2003
35. [JCP JSR 135.Mobile Media API2.0](#) 2003
36. [JCP JSR 177.Security and Trusted API for J2ME](#) 2004
37. [JCP JSR 179.Location API for J2ME](#) 2003
38. [JCP JSR 172.J2ME Web Service Specification](#) 2003

本文读者也读过(1条)

1. [冒海霞 移动通信安全中间件架构的设计和实现](#)[学位论文]2006

引证文献(12条)

1. [黎世红,程耕国,彭正涛 基于Android平台的Camera系统研究及应用](#)[期刊论文]-[电子设计工程](#) 2013(6)
2. [刘正,陈强 基于Android的GPS应用中的研究与开发](#)[期刊论文]-[中国西部科技](#) 2010(32)
3. [吴立勇,丁作文 基于Android系统振动测试仪研究](#)[期刊论文]-[工业控制计算机](#) 2013(12)
4. [陈勇,刘玉洁 基于J2ME手机个人助理系统的设计与实现](#)[期刊论文]-[电脑知识与技术](#) 2011(7)
5. [李杨,冯刚,李亮,罗拥华 基于Android的多媒体应用开发与研究](#)[期刊论文]-[计算机与现代化](#) 2011(4)
6. [陈勇,张曦焯 基于GPRS通信的J2ME家长信息系统](#)[期刊论文]-[计算机工程与设计](#) 2008(8)
7. [李培培,朱维杰 基于Android车载虚拟仪表人机界面设计](#)[期刊论文]-[电子科技](#) 2013(2)
8. [白璐,李国民 Android系统WMA文件播放功能的设计与实现](#)[期刊论文]-[电子科技](#) 2011(9)
9. [张仕成 基于Google Android平台的应用程序开发与研究](#)[期刊论文]-[电脑知识与技术](#) 2009(28)
10. [赵亮,张维 基于Android技术的界面设计与研究](#)[期刊论文]-[电脑知识与技术](#) 2009(29)
11. [公磊,ZHOU Cong 基于Android的移动终端应用程序开发与研究](#)[期刊论文]-[计算机与现代化](#) 2008(8)
12. [陆阳,乔钢柱,谭瑛 基于Android的智能家居程序开发与研究](#)[期刊论文]-[物联网技术](#) 2014(3)

引用本文格式：[刘向辉 基于Java的移动终端中间件的研究与设计](#)[学位论文]硕士 2005