

中低端智能设备与计算机间近距离 无线移动交互方案

荣耀^{1, 2}, 李昕³

(1 南京师范大学 中北学院, 江苏 南京 210046)

2 南京航空航天大学 信息科学与技术学院, 江苏 南京 210016

3. 南京电力自动化研究院, 江苏 南京 210003)

[摘要] 为了创建通过中低端智能设备与计算机间近距离无线交互应用, 提出了 MagicCube 移动解决方案. 该方案包括 MagicCube Mobile Framework 和 Xion Mobile Service 两个组成部分. 前者提供了一种中低端智能设备与计算机间近距离无线交互框架, 并为开发人员提供近似 .NET WinForm 的开发环境. 后者则是建立在前者和 Web service 之上的丰富客户端服务, 可以将远程 HTTP 服务转化成基于 MagicCube Mobile Framework 的近距离无线交互服务. 该方案扩展了 Microsoft .NET Framework 在近距离无线交互方面的无缝计算能力, 并在企业级 Web 应用服务与近距离无线交互应用服务间架起了桥梁. 实验证明, 可以为移动商务、家庭娱乐等带来全新的体验.

[关键词] 中低端智能设备, 近距离无线交互, 无缝计算, Web 服务

[中图分类号] TP393.09 **[文献标识码]** A **[文章编号]** 1672-1292(2008)02-0062-06

Resolution of Wireless Mobile Interaction at Close Range Between Medium and Low Level Intelligent Equipment and Computer System

Rong Yao^{1, 2}, Li Xin³

(1 College of ZhongBei Nanjing Normal University, Nanjing 210046, China)

2 College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

3 Nanjing Automation Research Institute, Nanjing 210003, China)

Abstract To create applications of wireless interaction at close range between medium and low level intelligent equipment and computer system, MagicCube mobile resolution is presented which comprises MagicCube mobile framework and Xion mobile service. The former provides a framework for wireless interaction at close range between medium and low level intelligent equipment and computer system, and offers a development environment similar to .NET WinForm while the latter is a rich-client service based on the former and web service, and can transform remote HTTP service to wireless interaction at close range service based on MagicCube mobile framework. The resolution extends the seamless computing ability of wireless interaction at close range of Microsoft .NET Framework, and establishes a bridge between enterprise web service and wireless interaction at close range service. Experiments show that MagicCube mobile resolution can bring users novel experience of mobile commerce services and home entertainment.

Key words medium and low level intelligent equipment, wireless interaction at close range, seamless computing, web service

手机、PDA 等中低端智能设备在现代商务中已经得到了广泛使用, 但是其中大多数并未拥有强大的计算能力, 甚至无法浏览为个人电脑设计的基于 HTML 的 Web 应用. 目前大部分手机除了通过 GPRS、CDMA 访问 WAP 网站应用以外, 几乎无法与外界 Web 站点进行交互; 矛盾的是, 运行在智能设备上的客户端程序(如 J2ME 程序)又需要用户及时下载更新. 由于各种设备性能差异较大, 缺乏一个统一的运行库和运行平台, 因此很难开发面向多种智能设备的商业应用.

为了创建可以满足多用户同时访问的近距离无线交互式应用服务, 且该服务无需在智能设备上安装

收稿日期: 2007-10-20

通讯联系人: 荣耀, 讲师, 博士研究生, 研究方向: 软件架构和软件工程等. E-mail: royay@royaba.com

任何额外的软件或收缴任何通信费用,同时开发人员可以像开发传统的 Web 应用那样开发基于该服务的商用应用,本文设计并实现了一种中低端智能设备与计算机间近距离无线交互移动方案: MagicCube Mobile Resolution(MagicCube移动解决方案, MMR). 该方案主要包括两个组成部分: MagicCube Mobile Framework(MagicCube移动框架, MMF)和 Xion Mobile Service(Xion移动服务, XMS). MMF提供了一种中低端智能设备与计算机间近距离无线交互式应用框架,并为开发人员提供近似 WinForm 的开发环境,它拓展了 Microsoft.NET Framework 在近距离无线交互方面的无缝计算能力^[1]. XMS 则是一种基于 MMF 和 Web service^[2]的丰富客户端服务,可以将远程 HTTP 服务转化成基于 MMF 的近距离交互服务,从而在企业级 Web 应用服务与近距离无线交互应用服务间架起桥梁.

1 名词和技术背景

1.1 智能设备

本文涉及的智能设备均指支持红外、蓝牙^[3]或无线局域网等无线通讯功能的便携式计算设备,这些设备还应该支持 AT 指令集和 Accessory 子指令集. 目前符合此规范的智能设备有 Ericsson 以及 Sony Ericsson 公司生产的移动通信设备,如 Sony Ericsson T610 系列等.

1.2 终端设备

本文涉及的终端设备均指支持红外、蓝牙或无线局域网等无线通讯功能的计算机、信息家电等,此外这些终端设备还应安装有 Microsoft.NET Framework 2.0(或更高版本)和 MagicCube Mobile Framework

1.3 GSM 手机 AT 指令集

AT 指令集是当今 GSM 手机标准之一,用于与外界设备通信、传递数据. AT 指令集可以对手机进行查询,如查询手机厂家、型号、设备状态等;也可以对手机进行操作,如命令手机发送短信、拨出电话等. 例如,指令 AT+CGMI 可以用于查询手机生产厂家,指令 AT+CGSN 可以返回手机机身序列号. 通常 AT 指令通过如红外、蓝牙等串口传入智能设备.

1.4 Accessory 子指令集

Accessory 指令集是 AT 指令集的一个子指令集,它是由 Sony Ericsson 等手机生产商提供的一种简单的无状态用户界面协议. Accessory 子指令集采用 UTF8 编码,已经内置在 Ericsson 以及 Sony Ericsson 大部分移动通信产品中. 该指令集可以将发令设备识别为手机的一个附件(Accessory),并可以命令手机在屏幕上显示消息框、文本框、选择框、多选框、滚动条以及菜单等基本用户界面元素,并且当手机用户操作时可以返回用户输入. 例如,命令 AT*EAID=11,2,"Title","Caption",10,""可以显示一个文本框,当用户完成输入时,手机会返回"*EAID=11,"Text"到终端("Text"代表用户输入). 命令 AT*EASM="Title",2,1,3,"Menu1","Menu2","Menu3",1可以显示一个菜单,当用户选择"Menu2"时,手机返回"*EAMI 2".

1.5 Microsoft.NET 无缝计算服务

Microsoft.NET Framework 致力于无所不在的“无缝计算”. 在“无缝计算”世界里,人们可以在任何时间、任何地点从任何设备得到任何想要的信息. Microsoft 发布了运行在 Windows Mobile 系统上的.NET Framework 精简版,其他非 Windows Mobile 智能设备则可以利用 GPRS 等无线通信手段,通过访问基于 ASP.NET 的 WAP 站点进行交互、计算^[4].

2 MagicCube 移动方案的设计与实现

2.1 MagicCube 移动方案概述

如前述, MagicCube 移动方案包括 MMF 和 XMS 两个组成部分. MMF 用于创建运行于 Microsoft.NET Framework 上的近距离交互式应用,应用本身运行在装有 MMF 的计算设备上(如个人电脑、服务器或其它消费类信息家电产品),可以通过无线通信传递 AT 指令集实现与近距离范围内的一个或多个移动智能设备的交互应用. 智能设备将会把计算设备终端识别为一个附件(Accessory). 智能设备无需为访问某项应用而额外安装任何软件,它只负责对终端传送的 AT 指令请求做出响应. 框架封装了对智能设备的生产厂家、型号、机身序列号(全球唯一识别码)的访问功能,用来确认客户端智能设备的身份. 同时会为每一个

会话创建一个唯一的 Application 对象,用来获得当前会话状态,维持与智能设备间的一对多访问.开发人员开发基于 MMF 的无线近距离应用像开发 WinForm 那样简单,无需深入了解其中的消息循环机制与 AT 指令细节. XMS 是在 MMF 基础上开发的一种支持多用户同时访问的近距离交互分布式应用服务器,它与远程 Web 服务器间会架起 HTTP Web 服务,将从 Web 服务器上响应来的 Magic 界面描述语言页面解析成一个 MagicCube Mobile Accessory Form 界面,并用无线通信手段将界面翻译成 AT 指令传递给智能设备,再将智能设备上的用户反馈翻译成一个 HTTP 请求发回 Web 服务器,同时也可以将从 Web 服务器上下载的文件传给智能设备.无论 Web 服务器端使用何种技术(静态 Magic 页面、ASP、ASP.NET、JSP 等),经过 XMS 服务器翻译都可以生成相应的 MagicCube Mobile Accessory 界面,最终显示在客户的智能设备上,实现交互.

2.2 MagicCube Mobile Framework 架构

MagicCube Mobile Framework 被部署在装有 Microsoft .NET Framework 的计算设备中(如图 1 中 1 所示),该电脑具备红外、蓝牙、无线宽带等近距离无线访问能力.同样具有无线通信能力的智能设备(如图 1 中 2、3 所示)与个人电脑间构架起无线通信可信连接(Trusted Wireless Connection),它们通过 AT 指令集进行交互.一台装有 MMF 的计算设备可以同时无线接驳多台智能设备.在不支持 AT 命令的智能设备上(如 Pocket PC),还需要另外安装解析执行 Accessory 指令集的客户端软件或硬件.

2.3 XMS Mobile Service 架构

图 1 展示的是一个标准的基于 XMS Mobile Service 开发的分布式应用. XMS 服务器(如图 1 中 2 所示)部署在 HTTP 应用服务器(如图 1 中 1 所示)与智能设备(如图 1 中 3 所示)之间,通过 HTTP 服务(Magic 页面)和无线通信(AT 指令)使二者间实现交互.一台应用服务器可以连接多个 XMS 服务器,而一台 XMS 服务器又可以同时支持多个智能设备访问.当接到智能设备访问请求时, XMS 服务会为每个访问设备建立一个独立的 HTTP Client,用来与 HTTP 应用服务器保持连接并维持会话. HTTP 应用服务器则负责将 ASP.NET 应用程序通过 IIS 执行并输出为 Magic 页面.而 XMS 服务将传回的 Magic 页面交给 Magic Page Parser 处理,后者将其转换为一个 MagicCube Mobile Accessory Form 对象,MMF 框架再将此 Form 对象渲染(Render)成 AT 指令,并通过无线连接最终呈现在智能设备上.

2.4 MagicCube Mobile Framework 功能模块

MagicCube Mobile Framework 使用 Microsoft .NET Framework 作为平台,共有 3 个类库(如图 2 所示),自底向上分别为负责底层通讯的 MagicCube Mobile Communication 模块、负责 AT 指令执行和解析 AT 指令返回参数的 MagicCube Mobile AT 模块,以及负责维持界面状态、逻辑和处理消息循环的 MagicCube Mobile Accessory 模块.

MagicCube Mobile Communication 模块负责与智能设备建立无线连接.首先需要为每一种有效的无线连接实现 IConnection 接口,并进行相关注册,然后利用 ConnectionBuilder 工厂类通过 Build(string connectionString)方法创建此无线连接的一个 Connection 实例. MagicCube Mobile AT 模块中 ATCommand 类负责根据参数生成 AT 指令文本,并通过 Connection 传递到智能设备上执行.根据智能设备返回的文本信息,会产生一个相应的 ATCommandResult 对象.利用 ATCommand 和 ATCommandResult 可以方便地生成和解析 AT 指令集. MagicCube Mobile Accessory 是 MMF 的



图 1 MMR 应用架构示意

Fig.1 Architecture of MMR applications

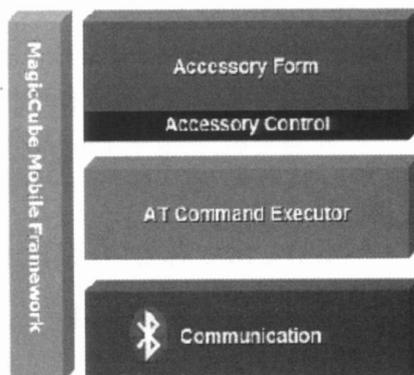


图 2 MMF 功能模块

Fig.2 Modules of MMF

Accessory Form 类库. 其中 Application 负责维持当前应用的会话和状态. MagicCube Mobile Accessory Control 是所有 Accessory 控件的基类, 它封装了一个简单消息循环机制, 消息体是 AT 指令, 消息分别使用 ATCommand 和 ATCommandResult 来发送和接收. 从 Control 派生下来的控件包括 Menu Form, Label TextBox, NumberBox, ListBox, CheckBox, SliderBar, MessageBox 等.

图 3 显示了 Accessory Form 模块中 Control 的生命周期和流程. 首先 Application 的 SetActiveControl (Control control) 方法自动调用 control 的 Show 方法, Show 方法会将该 control 控件设为当前的 Application.ActiveControl (MagicCube Mobile Accessory Control 类型), 并激发 Activating 事件. 紧接着 Show 方法会调用 Invalidate 方法, Invalidate 首先检查该控件是否是当前的 Application.ActiveControl (当调用 Show 方法时, control 总是当前的 Application.ActiveControl), 如果是则调用 Render(ATCommand command) 方法, 并触发 Rendering 事件; 反之则不做任何动作, 并结束 Show 方法的执行. Render 方法将 control 的属性、文本等信息呈现 (Render) 为 AT 指令, 并最终发送至智能设备解析, 所有继承自 MagicCube Mobile Accessory Control 的控件都必需重载 Render 方法来实现从 Accessory Form 控件到 AT 指令的转换.

图 4 揭示了控件如何接收智能设备反馈的 AT 指令消息并形成回送 (PostBack) 机制. 在当前 Application 的 Connection 连接接收到智能设备反馈的 AT 指令消息时, 会调用 OnResponded(ATCommandResult result) 方法, 参数 result 包含了经过解析后的消息信息, 然后 Application 会调用 Application.ActiveControl 的 RaisePostBackEvent(ATCommandResult result), 同时触发 PostBack 事件. 派生于 MagicCube Mobile Accessory Control 的控件重载 RaisePostBackEvent 方法, 对 ATCommandResult 进行分析, 然后触发控件自己的事件 (如 Changed, Submitting 以及 KeyDown/Up 等事件).

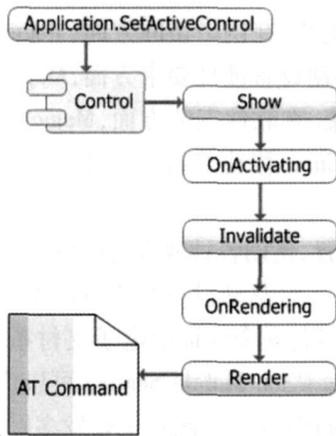


图 3 Accessory Control 的生命周期和工作流程
Fig.3 Life-cycle and work flow of Accessory Control

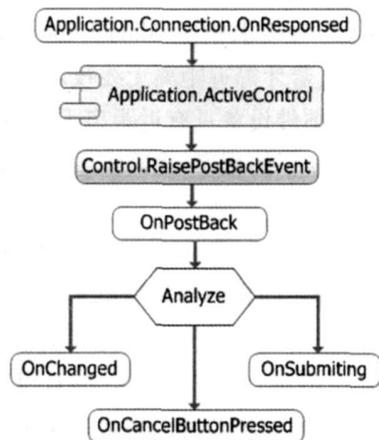


图 4 控件接收 AT 指令并形成 PostBack 的机制
Fig.4 Mechanism of receiving AT instruction and generating postback of control

2.5 X in Mobile Service 功能模块

X in Mobile Service 主要包括 Magic 引擎和 MagicCube Magic WebControls 两个模块. Magic 引擎是 X in 移动服务的内核. 当移动设备通过无线连接方式接入 X in 服务器后, Magic 引擎首先会对来访设备进行验证, 并将设备的相关信息 (如厂商、型号、机身序列号) 保存在 MobileSet 对象中. 然后 Magic 会分配一个进程对象 (MagicCube Magic Engine Process) 给来访设备. Process HttpAgent 会根据 MagicCube Magic Engine Server Config 中的设置向远程 HTTP 应用服务器的首页发出一个 GET 请求, 它将会调用 Process ProcessRequest 把服务器回应的 Magic 页面传递给 MagicCube Magic Engine PageParser 对象. MagicCube Magic Engine PageParser 将 Magic 页面载入 XmlDocument 对象, 通过 MenuParser, FormParser, MessageBoxParser 等解析器 (这些解析器均继承自 Parser 抽象类) 最终返回一个 MagicCube Magic Engine Page 类型的对象, 并将该对象赋予 Process Page Process Page 对象用 Page Control (通常是一个 MagicCube Mobile Accessory Form 容器) 包含当前 Magic 页面上所有表单元素 (Form Elements), 同时 Page 对象用 Page Parameters 集合保存这些表单元素的键和值 (Form Keys and Values). Process 在 Page 解析完成后会调用 Page Show 命令, 将 Page Control 通过 MMF 框架显示在智能设备上. 当使用智能设备的用户触

发某个事件时(如 Form 的表单提交事件, 或用户修改了文本框内的值触发了 TextBox 的 Changed 事件), Process 会命令 Page 执行 Submit 方法, Submit 方法先根据当前 Page Control 里所有子控件的值将 Page Parameters 更新, 再通过 HttpAgent 使用 POST 方法把 Page Parameters 以参数的形式提交到 HTTP 应用服务器, 然后 HttpAgent 再次调用 Process ProcessRequest 方法用新的 Magic 页面中的信息更新 Process Page 对象. 这样就完成了一次完整的消息循环, 实现了从远程 HTTP 应用服务器到智能设备上的信息交互.

MagicCube Magix WebControls 是为开发人员提供的 ASP.NET 服务器端 Magix 控件库, 该控件库提供了 Menu Form, Label TextBox, NumberBox, SliderBar, Trace 等服务端控件(所有控件均继承自 System.Web.WebControls WebControl), ASP.NET 开发人员可以使用这些控件构建自己的 Magic 页面. 同时还为每一个 Magix 控件提供了相应的 XSL 解析器, 可以把 Magix 语言解析转换成 HTML 代码, 使得开发人员可以在支持 XML, XSL 的 HTML 浏览器上调试 Magic 应用程序. 图 5 展示了在 MF 中如何实现传递控件事件.

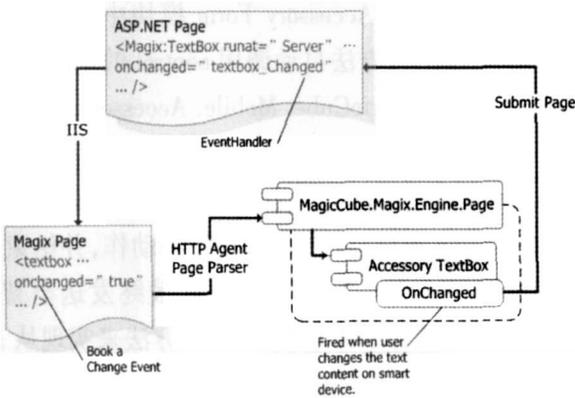


图 5 iXMS 中控件事件处理流程

Fig.5 Event handling of control in iXMS

3 MagicCube 移动方案的优势和应用前景

与 J2ME^[5]等技术相比, MagicCube 移动方案具有一些明显的优势. 在网络访问方面, MagicCube 移动方案使用红外、蓝牙等近距离无线技术, 无需额外的通信费用. 在移动设备硬件要求方面, MagicCube 移动方案仅仅要求硬件设备具有近距离无线访问能力并支持 AT 指令集. 在软件安装方面, MagicCube 移动方案客户端不驻留任何程序, 无需安装任何软件, 进一步而言, MagicCube 移动方案的软件更新在服务端完成, 移动设备端无需更新.

MagicCube Mobile Framework 可以用于开发交互式无线遥控装置, 使用各种兼容智能设备(如手机)充当信息家电设备(如家用娱乐型电脑、机顶盒等)的遥控器. 由于每一台移动设备在出场时都具有全球唯一识别码(或机身序列号), 因此还可开发在此基础上的带有数字签名安全认证的电子支付系统, 移动智能设备可以被识别为一张银行卡, 并且可以实现无线电子支付交易. Xion Mobile Service 可以广泛应用于商业领域中. 由于该服务可以将远程 Web 应用转换为本地近距离应用程序, 因此利于开发多网点式、多用户同时联机访问的商业咨询系统和电子商务系统, 例如蓝牙网络售货机、ATM 机、机场无线综合查询系统以及博物馆解说系统等.

4 应用实例

本文基于 MagicCube 移动方案开发了若干个原型应用, 包括 Magic Media Player 和 Magic Airport 航班查询系统等, 利用普通的手机获得了较理想的用户体验, 例证了 MagicCube 移动方案的优势.

4.1 Magic Media Player

Magic Media Player 是一个基于 Windows Media Player 控件的媒体播放器, 除了具有一般媒体播放器的功能外, 还可以利用智能设备进行遥控. 遥控不但可以控制播放器播放、暂停、导航与音量, 更可实时显示当前播放歌曲的信息和状态. 在该应用中使用了 MagicCube Mobile Framework 中的 Accessory Form, 其中包括 Menu Form, Label TextBox, SliderBar 以及 Icon 等控件, 还使用了 KeyDown, KeyUp 等事件捕捉键盘响应.

4.2 Magic Airport

Magic Airport 是一个 Xion Mobile Service 原型应用, 它是一个虚拟航班查询服务系统, 使用 ASP.NET 2.0+ SQL Server 2000 开发. 在 Magic Airport 中, 建立了 Bank Airport 两个 Web Service 将它们抽象为两个远程服务, 分别代表银行和机场的中央服务器. 在该服务系统运作过程中, 数据从 SQL Server 数据库到 Web Service, 然后到 HTTP 服务, 再经过 Xion Mobile Service 最终呈现在智能设备上, 实现了从底层数据库

到智能终端的通信和交互。

5 结语

MagicCube移动解决方案扩展了Microsoft.NET Framework在近距离无线交互方面的无缝计算能力,并在企业级Web应用服务与近距离无线交互应用服务间架起了桥梁,实验证明,可以为家庭娱乐、移动商务等领域带来全新的用户体验。接下来的研究工作包括提供对更多无线连接方式的支持、对更多智能设备的支持以及更丰富的客户端表现特性,实现更完善的身份识别验证系统以及功能更强的多层分布式MagicCenter中央服务器。将来还会将服务端延伸到Java技术平台,促使MagicCube移动方案成为胜任跨多种中低端智能设备、跨多种网络服务的移动计算平台。

[参考文献] (References)

- [1] David Chappell .NET大局观[M]. 2版. 荣耀,译. 北京: 电子工业出版社, 2006
David Chapell Understanding .NET[M]. 2nd ed Rong Yao Translated Beijing Electronics Industry Press 2006 (in Chinese)
- [2] 荣耀. ASP.NET 2.0实战起步[M]. 北京: 机械工业出版社, 2007.
Rong Yao Practical ASP.NET 2.0 [M]. Beijing China Machine Press 2007 (in Chinese)
- [3] David Kanmer, Gordon McNutt, Brian Senese, et al. 蓝牙应用开发指南——近程互联解决方案[M]. 李静, 奉继辉, 王婷, 等译. 北京: 科学出版社, 2003.
David Kanmer, Gordon McNutt, Brian Senese, et al. Bluetooth Application Developer's Guide The Short Range Interconnect Solution [M]. Li Jing, Feng Jhu, Wang Ting Translated Beijing Science Press 2003 (in Chinese)
- [4] 赵波, 安杨. 基于WAP的移动电子商务解决方案的研究与实现[J]. 计算机应用与软件, 2004, 21(10): 58-60
Zhao Bo, An Yang. Research of mobile electronic commerce resolution based on WAP [J]. Computer Applications and Software 2004 21(10): 58-60 (in Chinese)
- [5] 魏冬健, 高济. 基于J2ME平台的无线设备应用系统的开发[J]. 计算机应用研究, 2002(10): 146-148
Wei Dongjian, Gao Ji. Programming wireless devices applications based on the Java 2 Platform, Micro Edition [J]. Application Research of Computers 2002(10): 146-148 (in Chinese)

[责任编辑: 严海琳]