

基于 XML 的配置文件技术在 PDA 邮件代理系统中的实现

Implementation of Configure Files on PDA Email Agent System Based on XML Technique

施寒潇 (杭州 浙江工商大学计算机与信息工程学院 310035)

摘要:本文主要介绍了 PDA 电子邮件代理系统,重点阐述了 XML 技术在该系统用户配置文件中的应用。本文通过分析 PDA 在电子邮件方面现有的不足,首先提出了一个 PDA 邮件代理系统的设计框架及实现流程,最后详细阐述了如何运用 XML 技术具体设计与实现用户的配置文件,并对整个系统进行了客观的评价和展望。

关键词:XML 配置文件 邮件代理 个人数字助理

1 引言

随着 Internet 和 WEB 技术的发展,Email 已经在我们的日常生活中变的不可缺少。同时,伴随着移动技术的发展,人们对便携式移动信息处理设备的需求也日益迫切。PDA 的出现正迎合了这种需求,它的主要功能有日常记录、资料查阅,除此自外有些 PDA 设备还可以收发电子邮件,但是由于 PDA 软硬件设备的客观原因,它在接收邮件上存在很大的限制,如电子邮件的长度只能在 250 到 8000 字节之间,只能接收纯文本、不能接收附件等等。当 PDA 发现邮件大于自己所能接收的范围或者不能识别该种类型的邮件时,PDA 只能放弃该邮件的接收。这很大程度上限制了邮件服务在 PDA 上的应用和推广。针对此现状,本文提出了

PDA 邮件代理服务的系统框架,并且为使这种类型的代理更加灵活,适应不同类型 PDA 的 Email 要求,我们利用 XML 技术来进行相应封装。

2 PDA 邮件代理系统

2.1 系统功能

本系统是一个邮件代理系统,它的主要功能流程是 PDA 设备首先连接上邮件代理服务器,邮件代理服务器根据 PDA 用户提供的用户名、密码替代 PDA 与相应的 Email Server 建立连接,然后通过该代理接收邮件,最后根据邮件代理服务器中每个 PDA 用户的配置文件对用户的邮件进行相应的处理,再把处理过的邮件投递给 PDA 设备处理。

2.2 系统流程

图 1 即为系统的运行流程。具体的运行流程如下:(1) 用户向代理服务器发出 POP 请求,要求代理服务器向邮件服务器收信。(2) 代理服务器根据 PDA 用户提供的用户名和密码,替代用户向邮件服务器发出请求,索取指定的信件。(3) 代理服务器得到信件,把邮件保存在相应的接收缓冲区。(4) 根据数据库中指定用户的

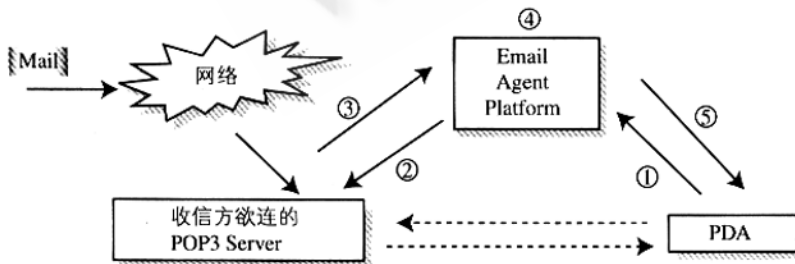


图 1 系统的运行流程

配置文件对收到的信件进行相应的处理。(5)代理服务器把处理好的邮件投递给用户。图中虚线表示没有邮件代理服务器的执行流程。

2.3 系统的开发与运行平台

本邮件代理采用 ANSI C 与 XML 语言开发,数据库为 MySQL,基于 Linux 平台。

运行本邮件代理,只需将邮件接收软件的“接收邮件服务器”改写为邮件代理服务器的域名或相应的 IP 地址,在“用户名”中填写“原用户名/邮件服务器的域名地址”。这样便完成了邮件代理服务器的设置。举例说明,帐户为 jim@163.com,代理服务器 IP 为 192.168.1.3,则只需将用户邮件接收软件中的“接收邮件服务器”改为 192.168.1.3;用户名改为 jim/pop3.163.com 即可使用本代理。

3 用户配置文件的设计与实现

3.1 配置文件的功

PDA 直接与邮件服务器进行交互时,当 PDA 发现邮件服务器中的某封邮件超过了接收大小范围,那么 PDA 将放弃该邮件的接收。但是总的来说,邮件正文很少超过 PDA 的接收范围,绝大多数情况是邮件的附件超过了 PDA 所能接收的范围(实际上现有的 PDA 设备一般只能处理很少类型的附件,一定意义上附件信息对 PDA 用户来说是多余的)。但如果放弃该邮件将导致用户所有信息的丢失。

针对上述情况,本邮件代理为每个 PDA 用户定义了配置文件。配置文件的功能主要体现在:用户通过设置配置文件来定义自己希望接收的邮件大小、邮件头、邮件体类型等信息。邮件代理将根据用户的配置文件对用户的信件进行处理,并把处理过的信件再投递给用户。所有用户的配置文件都以用户名为关键字段存储在数据库中,这样便于管理和查找。

3.2 配置文件的定义

在电子商务领域中,XML 作为数据交换的格式,已经有其不可替代的作用。但随着网络技术的发展,现在有一个新的趋势,就是使用 XML 作为配置文件的组织形式。尽管使用文本文件可以完成同样的功能,但是对于高度结构化和嵌套的数据,采用 XML 作为配置文件的组织形式,不仅使配置文件的可读性提高,而且使修改配置文件变的更加简单。

PDA 邮件代理在代理服务器中为每个 PDA 用户建立了一个配置文件,来确定用户想要接收的邮件大小与类型。配置文件的数据结构具体定义如下

```

: <xml version = "1.0" >
  <email >
    <emailsize >
      <size >2048 </size >
    </emailsize >
    <header >
      <return - path > n </return - path >
      <from > y </from >
      <to > y </to >
      <subject > y </subject >
      <date > y </date >
      <Message - ID > n </Message - ID >
      .....
    </header >
    <body >
      <text >
        <plain > y </plain >
        <html > y </html >
      </text >
      <image >
        <bmp > y </bmp >
        <jpeg > n </jpeg >
        <gif > y </gif >
      </image >
      .....
    </body >
  </email >

```

为了使描述简化,XML 文档只列出了部分的邮件头和邮件体的类型,实际上用户的配置文件应该包含所有的类型。

如果把上述 XML 文档用树状结构来表示,则表示如图 2 所示。

同样,为了描述简化,图 2 中只列出了部分邮件头和邮件体类型。

<emailsize> 节点下的叶子节点的字符值定义了 PDA 用户希望接收的邮件大小。邮件代理服务器将根据这个大小,从邮件中尽量取出完整信息给用户。这

里尽量取出完整信息分为下面二种情况:

(1) 如果邮件正文超过用户希望接收的邮件大小, 将把正文的一部分投递给 PDA 用户。

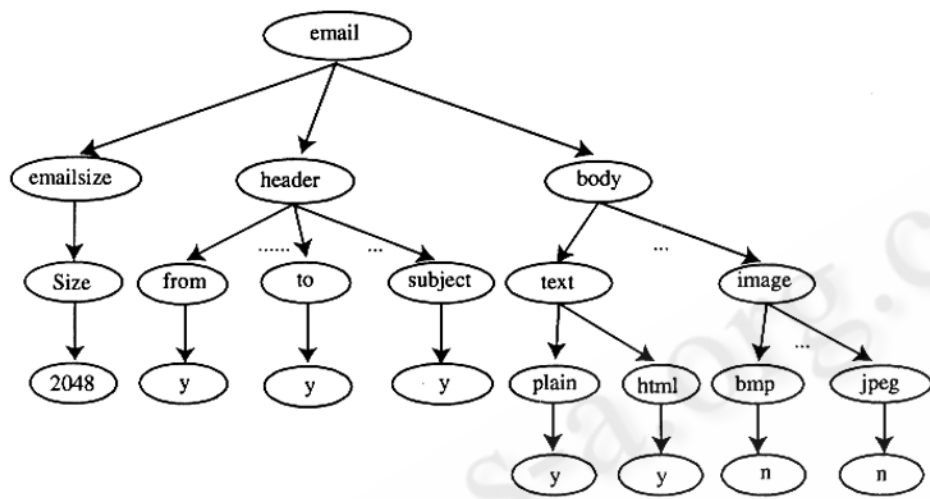


图 2 XML 文档结构的树形表示

(2) 如果附件超过用户希望接收的邮件大小, 邮件代理将放弃附件的接收。

<header> 节点下的子节点定义了各种邮件头。如果以 <header> 节点为祖先节点的某个叶子节点的字符值为 'y', 则该叶子节点的父节点是用户希望接收的邮件头。例如 <from>y</from> 表明用户希望接收“邮件发送者”这个邮件头。

<body> 节点下的子节点定义了邮件体类型。如果某个叶子节点的字符值为 'y', 则从 <body> 节点到该叶子节点所有经过的节点共同组成了邮件体的类型。子节点之间用 "/" 隔开。例如上述 XML 文档中, "text/plain" 是用户希望接收的邮件体类型。

有了上述的数据结构, 只能说明 XML 文档是“组织良好的”, 并不是“有效的”。一个有效的 XML 文档还需要 DTD 文档类型定义。但对于本邮件代理来说, 由于 XML 文档是系统管理员定义的, 所以总认为本 XML 文档是有效的。

3.3 配置文件的解析

定义了 XML 文档之后, 必须通过 XML 解析器来解析 XML 文档, 以便获得用户希望接收的邮件大小、邮件头与邮件体的类型。鉴于本邮件代理采用 ANSI C 语言开发, 系统使用了一个用 C 语言写的面向数据流的 XML 解析器 Expat。

对于不同的 XML 文档, Expat 解析器都按照如下流程执行解析:

(1) 利用解析器的 API XML_Parser - Create (const XML_Char * encoding) 创建 XML 解析器, 参数 encoding 可以为 US - ASCII、UTF - 8、UTF - 16、ISO - 8859 - 1, 也可以为空。如果参数 encoding 为空, 解析器将按照 XML 文档中的编码方式解析, 否则, 按照参数设置的编码方式进行解析。函数返回一个类型为 XML_Parser 的变量。

(2) Expat 解析器设置回调函数。对于图 2 所示的 XML 文档, 主要使用三个 API 设置回调函数。XML_SetElementHandler (p, start, end) 用来设置处理开始标记与结束标记的函数。XML_SetCharacterDataHandler (p, charHdlr) 用来设置处理开始标记与结束标记之间字符值的函数。之所以称为回调函数, 是因为编程人员不需要在程序中调用该函数, 解析器解析 XML 文档的时候会自动的调用已经设置的回调函数。例如上述的回调函数中, start 将用来处理开始标记, end 用来处理结束标记, charHdlr 将用来处理开始标记与结束标记之间字符值。为了各个回调函数之间共享数据, Expat 解析器还定义了 XML_SetUserData (p, &u), 该 API 用来将变量 u 设置为各个回调函数之间共享的数据(使用全局变量可以完成同样的功能, 不过由于全局变量在执行时一直占用系统资源以及通用性差, 本邮件代理采用 expat 解析器提供的 API)。

(3) 解析器调用 XML_GetBuffer (XML_Parser p, int len) 申请大小为 len 的缓冲区, 并将需要解析的 XML 文档读入缓冲区。

(4) 解析器调用 XML_ParseBuffer (XML_Parser p, int len, int isFinal) 解析由 XML_GetBuffer 申请的缓冲区中的 XML 文档, 由于 Expat 解析器可以读入 XML 文档的一部分到缓冲区中进行解析, 所以该 API 用最后一个参数来设置解析器解析的结束。如果 isFinal 为 0 时, 表明缓冲区中的 XML 文档是最后一部分, 这时 Ex-

pat 函数将停止解析。对于较小的 XML 文档可以设置 isFinal 恒等于 0,也就是一次把 XML 文档全部读入缓冲区。

本代理将通过解析用户配置文件,并按照“希望接收的邮件大小、希望接收的邮件头、希望接收的邮件体”的字符流顺序将解析结果写入缓冲区。最后邮件处理函数根据上述的解析结果从邮件源码中找出用户需要的内容,组合成一封新的电子邮件投递给代理平台。

由于解析的流程是固定的,所以这里主要介绍 XML_ParseBuffer 的执行过程和回调函数的实现。解析程序为了回调函数之间共享数据,定义了一个二维数组,并把二维数组的首地址赋给 XML_SetUserData(p, &u)的变量 u。程序是用二维数组来实现堆栈的功能。对读入缓冲区的 XML 文档来说,实际上是由开始标记、标记中的字符值、结束标记组成的字符流。

下面为 XML_ParseBuffer 的执行过程和回调函数的实现。

① 当解析器匹配到开始标记时,解析器自动调用开始标记处理函数。如直接调用 XML_SetElementHandler(p, start, end) 设置的 start 函数(可以为其他函数名,只要开始标记处理函数名与 XML_SetElementHandler 设置的参数名相同)。开始标记处理函数将开始标记 PUSH 堆栈。

② 当解析器匹配到标记之间的字符值时,与开始标记相同,解析器自动调用数据处理函数。数据处理函数首先判断堆栈栈顶标记是否为 size,如果是,则把标记之间的数据(用户希望接收的邮件大小)写入缓冲区,否则判断标记之间的字符值是否为“y”,如果是,则判断栈顶下一个标记是否为 header,如果是,把栈顶标记(用户希望接收的邮件头)写入缓冲区,否则按照“栈顶下一个标记/栈顶标记”写入缓冲区(由于邮件体类型由主类型/子类型组成)。

③ 当解析器匹配到结束标记时,解析器自动调用结束标记处理函数。由于该标记已经处理结束,结束标记处理函数将栈顶标记 POP 出堆栈。

(5) 当解析完 XML 文档后,调用 XML_ParseFree(XML_Parse p) 释放 XML 解析器所占的系统资源。

4 结束语

基于 XML 技术的配置文件很好的解决了用户个性化定制,使结构更加清晰。同时利用数据库技术对各用户配置文件的管理,极大地方便了程序的结构查询。虽然配置文件的设计只是本系统的其中一项工作,但它实现的好坏直接影响到用户的满意程度,这在某种意义上说它又是至关重要的。另外,从整个系统来看,由于现阶段的主客观原因导致本系统还存在着不足。如 PDA 接收邮件大小的限制并没有得到彻底的解决,这主要因为 PDA 自身的硬件条件,决定了 PDA 接收邮件的大小。针对这一情况,课题小组正在着手进行以下工作,在代理服务器中先对邮件进行分割,分割的大小为 PDA 所能接收的最大邮件大小,然后把一封电子邮件分割成几封电子邮件投递给用户。

参考文献

- 1 POP3 协议(RFC1939)[S]
- 2 Multipurpose Internet Mail Extension——MIME (RFC2015)[S]
- 3 W3C 关于 XML 的介绍[EB/OL].
<http://www.w3.org/XML>
- 4 耿刚、陈平,一种基于 XML 的 WEB 应用框架的设计[J],计算机工程与应用,2004(16):124—128。
- 5 吕腾、顾宁、闫萍,XML 文档的范式[J],小型微型计算机系统,2004(10):1836—1840。
- 6 陈丽、朱梦玲,局域网中邮件服务系统的构建和实施[J],核电子学与探测技术,2003(1):78—82。
- 7 刘宴兵、李春,使用 SMTP 实现邮件发送设计[J],计算机工程与应用,2002(8):142—144。
- 8 王庆波、方滨兴等,电子邮件过滤检测系统的设计与实现[J],计算机应用研究,2000(10):105—106。
- 9 Kevin Johnson. Internet Email 协议开发指南[M],北京机械工业出版社,2000。
- 10 H. M. Deitel, P. J. Deitel 等著,XML 编程技术大全[M],北京清华大学出版社,2002。