

清华大学图书馆流通通知短信发送服务的设计与实现^{*}

张蓓 周虹 张成昱 林容

(清华大学图书馆 北京 100084)

【摘要】介绍短信发送服务在清华大学图书馆流通领域的应用,详细说明短信内容提取子系统和短信发送子系统的设计和实现方案。

【关键词】短信服务 流通服务 内容提取

【分类号】G250.7

Design and Implementation of Short Message Service for Circulation Notification in Library of Tsinghua University

Zhang Bei Zhou Hong Zhang Chengyu Lin Rong

(Library of Tsinghua University, Beijing 100084, China)

【Abstract】This paper introduces the application of SMS in circulation field of Tsinghua University Library. The design and implementation of short message content extraction subsystem and short message transmission subsystem are expatiated

【Keywords】SMS (Short Message Service) Circulation service Content extraction

1 引言

图书馆向读者发送图书到期提醒、预约书到馆等流通知是现代图书馆流通服务的重要部分。图书馆在采用邮寄、电话通知和电子邮件方式递送流通知时,仍然存在通知送达时间不固定或不及时的情况,造成有些读者因未能及时接到通知而忘记按时归还图书,产生逾期罚款,或者失去借阅预约图书的机会。因此,寻找新的方式作为流通知发送的有益补充就成为提高流通服务质量所必须解决的问题。统计数据显示,中国现有手机用户已经超过 6 亿,每月短信发送量超过 500 亿条^[1],通过手机短信进行沟通和交流已经成为一种广为大众接受的方式。对于以教师和大学生为主体的高校图书馆用户而言,手机拥有率较高,因此,利用短信方式进行服务在高校图书馆是具有良好用户基础的^[2]。如果将短信方式作为向读者发送流通知的第 4 种方式,不但可以实现通知产生和发送基本自动进行,达到节省人力的目标,而且对读者而言这种途径更方便及时,进而可以提高图书流通知率,使图书馆的资源得到更加有效的利用^[3,4]。在这样的背景下,清华大学图书馆利用现有技术和资源,结合 NNOPAC 系统自动生成流通知的功能,设计并实现了流通知的短信发送服务。

收稿日期: 2008 - 11 - 21

收修改稿日期: 2008 - 12 - 04

^{*} 本文系教育部人文社会科学研究规划基金项目“基于无线广域网的移动数字图书馆实现和服务机制的若干关键问题”(项目编号: 08JA870010)的研究成果之一。

2 设计方案

2.1 设计思路

清华大学图书馆采用的图书馆管理系统是美国 INNOVATIVE公司的 NNOPAC系统,系统已有自动生成流通通知并根据读者记录中包含的 E-mail地址定时发送流通通知的功能。虽然该系统声称已有可供用户选用的短信发送流通通知模块,但需要另行购买,价格不菲,而且该模块是按照与美国移动运营商平台对接设计的,并不能完全保证在中国适用,因此本馆决定结合中国移动提供的“企信通”业务另行开发短信发送流通通知的功能。

根据 NNOPAC系统的流通通知发送原理,流通通知不会自动发送给没有登记 E-mail地址的读者,但可以按照一定格式打印出来供图书馆邮寄或者由馆员根据通知中显示的电话号码电话通知读者。也就是说,NNOPAC系统能提供一个具有一定格式的、除已通过 E-mail方式发送的流通通知以外的所有流通通知文本。要通过移动的“企信通”平台发送流通通知,需要提供接收者的手机号码列表以及通知文本内容。由于每天的接收者不固定,因此借鉴 NNOPAC系统通过 E-mail发送流通通知的方法,需要将读者用于接收通知的手机号码放入读者记录中特定的字段并且显示在生成的流通通知文本中,这样就可以从生成的通知内容中获取接收者的手机号码。此外,由于 NNOPAC系统生成的流通通知内容过于冗长,不适用于短信发送,而且每日发送的通知类型不止一种,因此还需要将通知内容按照通知类型进行转化处理,并与接收者的手机号码进行匹配,以产生符合“企信通”平台要求的发送队列。

2.2 实施步骤

首先,本馆向读者发放了解用户是否需要短信发送流通通知的意见调查表,调查结果显示,读者对这种方式很感兴趣,特别是上网不太方便的本科生以及工作事务比较繁忙的教职员工,大多数读者都愿意留下自己的手机号码,接受图书馆这项短信服务。

在此基础上,结合 NNOPAC系统中 OPAC模块用于读者自行维护个人信息的功能,引导读者在网上注册本人的手机号码信息。由于自行维护读者数据是读者自愿进行的,这就保证了注册手机号码的读者认同

短信接收流通通知这种方式,而不会认为图书馆发送的信息是垃圾信息。同时,读者可以在系统中随时根据需要修改或者删除本人的手机号码,这不仅确保了号码的及时更新,而且与另行建立用户库相比,避免了数据的不一致,大大减少了图书馆对手机号码维护的成本。

在搜集读者手机号码的同时,还需要对 NNOPAC系统产生的流通通知文本格式进行一定修改,使通知文本中包含用于发送的手机号码,以便后续程序分离出用于发送的号码和文本互相匹配的队列。最后,实现用于提取短信内容的程序以及设计用于发送短信的模板,设计工作流程,最终实现以短信方式发送流通通知。

2.3 系统架构

基于上述实施步骤,流通通知短信发送服务的总体设计架构如图 1所示,可以划分为 4个环节:

(1) 手机号码的采集:基于 NNOPAC系统的 OPAC模块实现读者手机号码的采集。

(2) 通知信息的生成:NNOPAC系统定时按照规定格式生成“到期提醒”、“预约催还”、“过期通知”和“预约取书”4类通知的文本文件,保存至服务器。

(3) 短信内容的提取:短信内容提取子系统将对流通通知的文本文件进行解析,分析关键信息,如读者姓名、手机号码、通知类型、书名等,并将这些信息按照指定的格式保存为短信发送任务列表。

(4) 短信的发送:短信发送子系统读取短信发送任务列表,按照定制的模板格式将短信发送至指定的读者。

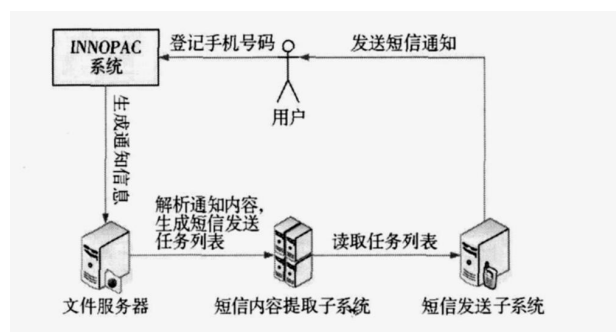


图 1 流通通知短信发送服务的总体架构示意图

在上述环节中,短信的内容提取和发送属于该服务的关键环节,下面将重点介绍其实现过程。

3 系统实现

3.1 短信内容提取子系统

(1) 通知文本及任务列表格式

短信内容提取子系统的功能是解析 NNOPAC 系统生成的 4 类流通通知的文本文件,并生成待发送短信列表文件,供短信发送子系统使用。4 类通知文本的格式是类似的,每个通知文本都由具有固定格式的多条记录组成,以预约取书通知中的一条记录为例,文本格式如图 2 所示。

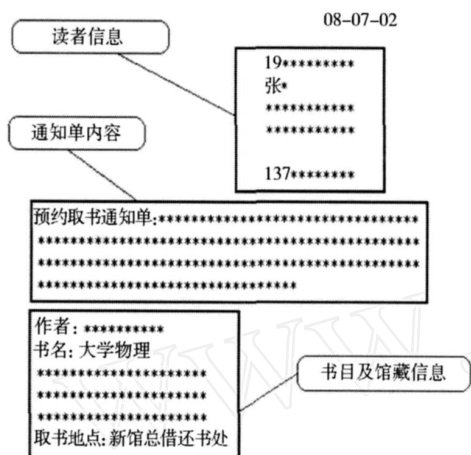


图 2 预约取书通知文本样例

每条通知记录以日期开始,其后包括“读者信息”、“通知单内容”和“书目及馆藏信息”三部分。对于预约取书通知而言,需要从中提取出读者的姓名、手机号码、预约书的书名以及取书地点,然后将提取出的信息加入到预约取书通知的任务列表,提取出的信息在任务列表中的格式如下:

张*,137*****,大学物理,新馆总借还书处

(2) 具体实现

短信内容提取子系统采用 Java 作为开发语言,通过 Apache Commons 软件包中的 org.apache.commons.io.FileUtils 类来读取文本文件^[5],实现内容提取。处理通知文本文件的基本步骤是:

寻找 yy-mm-dd 形式的日期标志,作为每条通知记录的开始点;

向下移动若干行,定位至学号、姓名等信息所在行,读取相关信息;

将读取的信息保存在 List 类型的数据结构中,同时

剔除重复数据;

将 List 中保存的信息输出到文本文件,即生成待发送短信的任务列表文件。

以下是关键的实现代码。

```

List list = new ArrayList(); // List 数据结构,用于保存任务信息
...
LineIterator it = null;
try {
    it = FileUtils.lineIterator(new File(NoticeUtil.getPickupNoticePath()));
    while (it.hasNext()) {
        String line = it.nextLine();
        // 记录开始位置
        if (NoticeUtil.matchShortDate(line)) { // 匹配日期格式 yy-mm-dd
            // 前进 2 行,获取学号
            String id = NoticeUtil.jumpIterator(it, 2).trim();
            // 前进 1 行,获取姓名
            String name = it.nextLine().trim();
            if (name.length() == 0) name = "未知名";
            // 前进 4 行,获取电话号码
            String phoneno = NoticeUtil.jumpIterator(it, 4).trim();
            // 匹配手机号码
            String cellphoneno = NoticeUtil.matchCellPhoneNo(phoneno);
            // 书名
            String bookName = NoticeUtil.jumpIterator(it, 8);
            bookName = bookName.replaceAll("TITLE:", "").trim().replaceAll("(\\s)\\s*", " $1");
            if (bookName.length() > 20) bookName = bookName.substring(0, 20);
            // 无手机号码记录的处理
            if (cellphoneno == null) {
                ...
            } else {
                // 取书地点
                String locationName = NoticeUtil.jumpIterator(it, 4);
                int t = locationName.length();
                locationName = locationName.substring(15, t);
                // 输出行
                String ouLine = name + "," + cellphoneno + "," + bookName + "," + locationName;
                // 只输出有电话号码的记录,且不重复
                NoticeUtil.add2List(list, ouLine);
            }
        }
    }
}
    
```

```

}
//输出文件
NoticeUtil writeList2File (NoticeUtil getPickupNoticeOutputPath
(), list);
...
}
...

```

需要说明的是,为了尽可能地保证向所有读者发送流通通知,短信内容提取子系统除了生成待发送短信的任务列表文件外,还对无手机号码的记录进行特别提取处理,后续经人工电话通知。

其他三种类型的通知文本也采用了类似的处理方法,区别在于读取信息的位置和在任务列表中输出的内容存在细微的差别。

3.2 短信发送子系统

短信发送子系统的功能是从短信内容提取子系统中获取待发送短信的列表文件,然后向读者发送流通通知。清华大学图书馆采用中国移动提供的信息机进行短信发送,信息机通过互联网与电信运营商的统一通信服务器连接,图书馆员通过信息机上预装的“企信通”平台向读者发送流通通知短信。短信发送子系统的系统架构如图 3 所示。

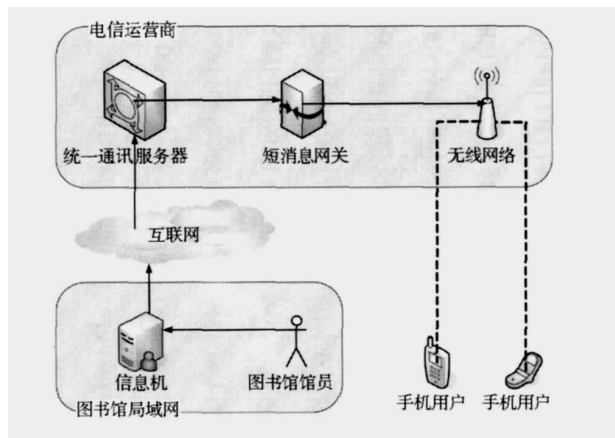


图 3 短信发送子系统系统架构图

采用“企信通”平台发送短信的关键是模板业务的定制。普通的短信发送多为一对一服务或者群发内容相同的短信。如果需要批量发送短信,并且针对不同用户的短信内容不同,使用普通的短信发送方式显然无法满足要求,这就需要用到模板业务来实现既定的功能。所谓模板业务就是根据实际需求,事先定制好短信文本的样式。这里的短信文本不再只是静态内

容,而是可以与后台数据库中的数据结构建立映射关系,实现信息的动态化。

以预约取书通知为例,具体的模板业务定制过程如下:

(1) 为新模板业务选择业务类型,这里选择“发送类”,即该业务是用于发送短信的。

(2) 填写业务名称、业务说明、业务代码及维护用户等基本信息。

(3) 将该业务与指定数据库进行连接,可以是本地数据库,也可以是远程数据库。这一步骤确定了业务的数据源格式,例如预约取书通知业务的数据源格式如表 1 所示。

表 1 预约取书通知业务的数据源格式

| 字段名称 | 字段说明 | 字段长度 | 字段类型 |
|----------|------|------|------|
| name | 姓名 | 10 | 文本 |
| number | 手机号码 | 11 | 数字 |
| book | 书名 | 40 | 文本 |
| location | 取书地点 | 20 | 文本 |

(4) 设置该业务的发送对象,这里的发送对象指定为当前业务连接的数据库中的 number 字段。

(5) 设置短信发送内容的格式,如图 4 所示:



图 4 预约取书通知业务的短信发送内容格式

短信发送内容的格式设置好以后,一个过期通知业务就建立成功了。流通部在发送预约取书通知时,首先从短信内容提取子系统中获得预约取书通知的任务列表,该任务列表中的记录格式形同“name, number, book, location”,符合模板业务的源数据格式,数据会自动嵌入短信发送文本中。短信发送子系统将会把动态生成的短信发送到出现在任务列表中的读者手机上。

4 结 语

利用短信发送流通通知的服务是清华大学图书馆对于无线移动通信技术在图书馆中应用的有益尝试。该服务一经推出,因为其接收方便,读者获得告知及时,得到了本馆读者的积极支持和响应,现在已有一万多名读者在本馆注册了手机号码,用于接收各类流通

通知。短信通知因其方便、快捷的特点提高了图书馆的服务工作质量。今后,我馆将进一步利用短信服务平台拓展与读者的互动途径,使图书馆的服务更加人性化 and 多元化。

参考文献:

- [1] 2008 年 8 月电信业统计月报 [EB/OL]. [2008 - 10 - 31].
http://www.miit.gov.cn/n11293472/n11293832/n11294132/n11302706/11518197.html

- [2] 周维彬,董成娣. 关于开展图书馆短信通讯服务的探讨 [J]. *图书馆界*, 2004 (2): 37 - 40.
[3] 刘云鹏. 手机短信:图书馆管理与服务的新方式 [J]. *国家图书馆学刊*, 2006, 56 (2): 59 - 60.
[4] 王景艳. 图书馆书情短信系统设计 [J]. *农业图书情报学刊*, 2008, 20 (1): 37 - 39.
[5] Apache Software Foundation: Apache Commons D 软件包用户指南 [EB/OL]. [2008 - 10 - 12]. http://commons.apache.org/iodescription.html

(作者 E-mail: zhangbei@lib.tsinghua.edu.cn)

OCLC 将为瑞士的元数据中心提供开发工具

2008 年 12 月 10 日,在德国的 Oberhaching, OCLC 同意开始实施与 SwissBib 达成的相关协议。SwissBib 是瑞士各个大学图书馆和瑞士国家图书馆联合推出的基于“Library 2.0”的元目录系统,这个元数据交换中心将为参与其中的图书馆的联合馆藏提供整理和发布平台。SwissBib 利用最新的搜索技术,为那些研究瑞士学术图书馆馆藏的终端用户提供一个优良的环境。该项目将从 2008 年 11 月开始,到 2010 年 2 月完成。

为了实现目标,SwissBib 的遴选委员会选择了 OCLC 提供的组合方案:利用 OCLC 的中央书目系统软件 (Central Bibliographic System, CBS) 作为后端书目组件处理其大约 1 700 万条记录,从图书馆目录中获得的所有记录都被导入到 CBS 系统中,经过转换、匹配及完善,最后转入发布平台, OCLC 为该平台提供了 TouchPoint 发现功能,并集成了 FAST 索引引擎。

SwissBib 选拔委员会寻求到了一种最为简洁有效开发方案来处理目录,为瑞士的科学信息提供方便、快捷及广泛的获取途径,项目预期的终极目标是取代目前正在瑞士各个大学图书馆使用的 OPAC 系统的绝大部分功能。该解决方案将通过瑞士书目数据与 WorldCat 的同步,来提高瑞士各个大学图书馆在互联网全球读者视野中的能见度。

SwissBib 的项目负责人 Hannes Hug 先生说:“OCLC 的元数据中心为 SwissBib 提供一个灵活有效的系统方案,这将大大降低瑞士图书馆‘生态系统’的复杂性。与开发系统的结合使得我们有可能为用户切入点提供一个现代化的接口,以帮助他们获取瑞士大部分图书馆的资源。”

TouchPoint 的全球产品经理 Marianne Klomp 说:“通过 SwissBib 项目我们可以展示元数据管理工具 CBS 的灵活准确的一体化功能,它用 FAST 作为搜索引擎,用 TouchPoint 作为必须的终端用户接口。通过这种方法使一个有效的后台工具与 Web 2.0 的前端相结合,使得图书馆全体馆员及所有终端用户都能得到最好的服务”。

SwissBib 项目将在多家单位的参与下共同完成,包括:瑞士西部图书馆网络,瑞士德语区图书馆网络,瑞士国家图书馆,瑞士学术图书馆联盟及巴塞尔大学图书馆赞助的 ETH - Bibliothek。SwissBib 又被称作 E-lib-ch,是瑞士各个大学合作的总体创新项目中的一个,由瑞士大学联合会赞助。更多信息请访问 <http://www.swissbib.org>

(本刊讯)