

# 一个移动应用个性化集成框架的研究 及其在 Android 平台的实现

张栋栋 徐 锋

(南京大学计算机软件新技术国家重点实验室 南京 210046)

(南京大学计算机科学与技术系 南京 210046)

**摘要** 随着互联网和智能手机的日益普及,移动应用数量呈现爆炸式增长,海量的移动应用既是机遇也是挑战。从开发者角度看,基于大量的已有移动应用快捷构造新的移动应用成为了可能,但当前大部分的开发工具对移动应用集成的支持仅停留在应用编程接口和底层运行机制上,尚未出现更高层次的面向执行流程的集成支持;从用户角度看,从众多移动应用中选择符合自己个性化需求的应用成为了可能,但目前大量的移动应用推荐系统集中在单个应用的个性化推荐问题上,未见面向用户个性化需求的移动应用执行序列推荐方法。为此,提出一个移动应用个性化集成框架,主要包括:1)定义了一套意图流程描述执行语言,以便开发者从更为自然的执行流程角度完成新移动应用的构造;2)给出了一个移动应用序列偏好度预测算法,用于解决移动应用执行序列的个性化推荐问题。在当前典型的移动应用平台 Android 上,实现了相应的移动应用个性化集成开发工具和运行支撑机制,并通过实例初步验证了上述方法的合理性。

**关键词** 移动应用,集成框架,序列推荐

**中图分类号** TP311.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.11.013

## Personalized Integration Framework for Mobile Applications and its Implementation on Android Platform

ZHANG Dong-dong XU Feng

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China)

(Department of Computer Science and Technology, Nanjing University, Nanjing 210046, China)

**Abstract** The explosive growth of mobile applications has brought new opportunities and challenges to both developers and users. From the perspective of developers, it is now possible to build new mobile applications rapidly based on massive existing mobile applications. However, such high-level integration is largely ignored by most existing developing tools whose focuses are on the API level. From the perspective of users, while recommending a single personalized application for users has been widely explored, recommending a whole application integration such as application sequences still remains as an open problem. We proposed a personalized integration framework for mobile applications, and our framework contains two major parts: 1) defines an intent process execution language, which could facilitate the developers to build applications more naturally, 2) gives the algorithm to evaluate the preferences of different mobile application sequences, which could support personalized recommendation of application sequences. To verify the rationality of the framework, we implemented it as a developing tool and runtime supporting mechanism on the Android platform.

**Keywords** Mobile application, Integration framework, Sequence recommender

## 1 引言

随着互联网和智能手机的日益普及,移动应用数量呈现爆炸式增长。截止 2012 年 10 月份,苹果 App Store 和 Google Play 移动应用数量均超过 70 万。App Store 平台应用下载总数超过 500 亿次,Google Play 平台应用下载总数超过 480 亿次。海量存在的移动应用给人们带来了新的机遇和挑战。

从开发者角度看,基于大量已有的移动应用快捷构造新的应用成为了可能。在开发过程中,若能将已有的成熟移动应用集成进来,无疑会大大缩短开发周期,降低开发风险,同时保证产品质量。但当前大部分移动应用开发工具对应用集成的支持仅停留在应用编程接口和底层运行机制部分,尚未出现更高层次的面向执行流程的集成支持。以当前典型的移动应用平台 Android 为例,虽然提供了意图(Intent)机制用于不同应用间相互调度和交换数据<sup>[1]</sup>,但这种机制对移动应用

到稿日期:2013-09-16 返修日期:2013-11-18 本文受国家 863 计划(2012AA011205),国家自然科学基金(青年基金项目)(61100038),国家自然科学基金(中美软件合作研究项目)(61361120097),国家基金委可信集成项目(91318301)资助。

张栋栋(1989-),男,硕士,主要研究方向为可信计算、推荐系统,E-mail:ddzhangnju@163.com;徐 锋(1975-),男,博士,教授,博士生导师,CCF 会员,主要研究方向为系统安全、网络安全和可信计算。

集成的支持仅停留在应用编程接口层面,因而带来了两个问题:1)新应用请求是开发者硬编码得到的,在发起该应用请求时,会弹出一个候选应用对话框,用户每次都需要从候选列表中点选应用,虽然可以设置默认应用,但难以构造一个统一的方法来智能地过滤候选应用列表;2)限制开发者对更多的应用进行序列化集成以完成更强大的功能,以下述场景为例,某开发者希望完成一个很常见的功能:首先获取一幅图片,接着进行图片美化,最后将编辑后的图片分享出去。这个功能中的每一步均有大量成熟的移动应用可以利用,拍照类和文件浏览类应用可以获取图片,图片编辑类应用可以处理图片,微博等社交化应用可用于分享。开发者不希望进行重复开发,期望能直接调用一个已存在应用构成的序列完成上述流程,即移动应用的“序列化复用”,现有的移动应用开发工具无法实现类似功能。

从用户角度来看,海量存在的移动应用使得选取符合自己个性化需求的应用成为可能,移动应用推荐系统通过获取用户、上下文、应用之间的潜在关联,为用户进行个性化应用推荐<sup>[2]</sup>。Woerndl等<sup>[3]</sup>将位置等上下文信息引入移动应用推荐系统,基于其他用户在相同上下文环境下的安装信息为新用户进行推荐。Zheng Yan等<sup>[4]</sup>提出了基于信任行为的声誉和推荐系统 TruBeRepec,在大量用户调查的基础上建立了一个信任行为模型,提出算法预测用户对移动应用的信任程度,进而进行应用推荐。文献<sup>[5]</sup>介绍了世界上最大的免费应用商店 GetJar 在移动应用推荐系统中如何解决稀疏数据集问题。Bo Yan等<sup>[6]</sup>基于 Android 平台开发了 AppJoy,通过收集用户的详细使用记录,利用基于物品的协同过滤算法为用户进行移动应用推荐。Christoffer等<sup>[7]</sup>指出,移动应用推荐系统中常常采用的协同过滤和基于内容的过滤,都面临“新用户”的问题,它们通过引入隐式反馈和上下文感知,较好地解决了冷启动问题。可以看到,目前大量的移动应用推荐系统集中在单个移动应用的个性化推荐问题上,未见面向用户个性化需求的移动应用执行序列推荐方法。在移动应用执行序列环境下,用户执行过程中的每一步都要从“同斥”的应用列表中选取符合自己喜好的应用,这无疑浪费了大量时间,若选取到不理想的应用,用户还有可能停止该动作序列,甚至卸载应用,因此如何得到一个理想的移动应用执行序列是亟待解决的问题。

为此,我们提出一个移动应用个性化集成框架来解决开发者和用户面临的问题,主要包括:1)定义了一套意图流程描述执行语言,方便开发者从更为自然的执行流程角度完成新移动应用的构造;2)给出了一个移动应用序列偏好度预测算法,用于解决移动应用序列的个性化推荐问题,同一功能在不同用户的手机上会通过更适合自己的移动应用序列来执行,这种机制可以大大提升用户体验。

## 2 移动应用个性化集成框架

为解决开发者和用户面临的问题,我们提出了一个移动应用个性化集成框架,如图 1 所示。

开发者在进行移动应用序列化集成时,首先需要对动作序列进行描述。我们提出意图过程执行语言 IPEL(Intent Process Execution Language)来描述动作序列。

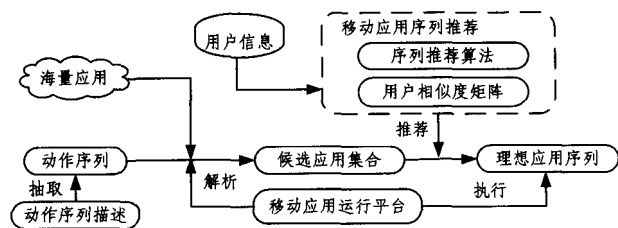


图 1 移动应用个性化集成框架

得到动作序列后,海量存在的应用使得每个动作都有多个候选应用可以选择,而移动应用运行平台提供了类库来获取这些候选应用集合。

同时,该框架包含一个应用序列推荐模块。基于用户相似度矩阵和序列推荐算法,该模块利用候选应用集合得到一个理想的应用序列推荐给用户,作为最终的应用执行序列。

### 2.1 相关定义

为准确地描述框架,下面对框架中包含的概念进行结构化定义。

**定义 1(动作)** 定义  $A$  为动作,它是形如  $\langle action, \langle type, uri \rangle, category, \langle key, value \rangle \rangle$  的数据结构,其中  $action$  指定动作名称,  $\langle type, uri \rangle$  指定请求的数据类型和数据源,  $category$  指定请求的类别,  $\langle key, value \rangle$  指定请求时携带的附加信息。动作由大写字母表示,代表发起某种类型的应用请求。

**定义 2(动作序列)** 动作序列是形如  $\langle X, Y, Z, \dots \rangle$  的顺序序列,其中  $X, Y, Z$  要么是某个动作  $A$ , 要么是某些动作的并  $A \cup B \cup \dots$ 。

**定义 3(应用)** 定义  $a$  为应用,它是形如  $\langle A, \langle package, class \rangle \rangle$  的数据结构,其中  $A$  为一个动作,  $\langle package, class \rangle$  为能响应  $A$  的具体移动应用的组件信息,  $package$  对应于移动应用的包名,  $class$  对应于组件的类名。

应用由小写字母表示,代表了一个具体的移动应用,例如新浪微博可以响应分享图片请求,形如:

$\langle A, \langle "com.sina.weibo", "com.sina.weibo.EditActivity" \rangle \rangle$

其中,  $A$  为发起图片分享类型应用请求的动作。

**定义 4(应用序列)** 定义  $\alpha$  为应用序列,它是形如  $\langle a, b, c, \dots \rangle$  的移动应用构成的顺序序列,  $a_i$  表示第  $i$  个移动应用,  $\alpha.size$  表示序列长度。

### 2.2 意图过程执行语言

为了便于开发者对动作序列进行描述,需要定义一套动作序列描述语言。我们提出意图过程执行语言 IPEL 来描述动作序列,该语言采用“松耦合”思想,即语言中并不指定调用的具体移动应用,而是在平台运行时时刻绑定到已安装的应用,这种策略使应用序列推荐成为可能。

IPEL 基于可扩展标记语言 XML( eXtensible Markup Language)<sup>[8]</sup>,语法结构由 XSD(XML Schema Definition)<sup>[9-11]</sup>约束。

首先对用到的属性进行定义:

```
<xs:attribute name="type" type="xs:string"/>
<xs:attribute name="uri" type="xs:string"/>
<xs:attribute name="key" type="xs:string"/>
<xs:attribute name="value" type="xs:string"/>
<xs:attribute name="processname" type="xs:string"/>
<xs:attribute name="actionname" type="xs:string"/>
```

```

<xs:attribute name="categoryname" type="xs:string"/>
<xs:attribute name="describe" type="xs:string"/>
<xs:attribute name="skip" type="xs:boolean"/>
<xs:attribute name="reselect" type="xs:boolean"/>

```

下面定义复合元素,首先定义表示请求动作的(*intent*)元素。该元素包含(*action*)、(*category*)、(*data*)、(*extra*)4个子元素,对应于2.1节中对“动作”数据结构的定义。(*action*)元素的*name*属性为请求的动作名称,(*category*)包含了响应该动作的组件的类别信息,(*data*)元素包含需要传递的数据信息,其中*type*属性标记数据的MIME(多用途互联网邮件扩展,Multipurpose Internet Mail Extensions)类型,*uri*属性标记数据来源地址,(*extra*)元素包含动作携带的附加信息。(*intent*)元素的完整定义如下:

```

<xs:element name="intent">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="action">
        <xs:complexType>
          <xs:attribute ref="actionname" use="required"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="category" minOccurs="0">
        <xs:complexType>
          <xs:attribute ref="categoryname"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="data" minOccurs="0">
        <xs:complexType>
          <xs:attribute ref="type"/>
          <xs:attribute ref="uri"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="extra" maxOccurs="unbounded">
        <xs:complexType>
          <xs:attribute ref="key"/>
          <xs:attribute ref="value"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

有时开发者希望某个功能可以由多个不同类型的动作完成,例如获取一幅图片,该功能可以通过拍照或浏览文件两种不同类型的动作完成,定义(*pick*)元素描述此类请求。(*pick*)元素拥有多个(*intent*)子元素,每个子元素代表了一个请求动作。(*invoke*)元素则表示该功能只由一个固定的动作完成。这两个复合元素的定义如下:

```

<xs:element name="pick">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="intent" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="invoke">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element ref="intent"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

最后定义完整的语法结构。由于手机平台的特殊性(同一时间只能和一个应用交互),IPEL只考虑了顺序类型的功能流程,没有考虑并发等其他类型。流程从(*process*)元素开始,该元素用于唯一的标识流程;(*request*)元素表示一次功能请求,*describe*属性用于描述该功能请求,例如“获取图片”,*skip*属性用于标识该步是否可以跳过,*reselect*属性用于标识该步是否可以重新选择。每次功能请求是通过(*pick*)或(*invoke*)元素描述的。完整的语法结构如下:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="process">
    <xs:complexType>
      <xs:attribute ref="processname" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="request" maxOccurs="unbounded">
    <xs:complexType>
      <xs:attribute ref="describe"/>
      <xs:attribute ref="skip"/>
      <xs:attribute ref="reselect"/>
    </xs:complexType>
  </xs:element>
  <xs:element ref="pick"/>
  <xs:element ref="invoke"/>
</xs:schema>

```

## 2.3 移动应用序列推荐

开发者定义了动作序列后,用户每执行一步都要从候选应用集合中选取应用,这种方式一方面浪费用户时间,另一方面容易带来糟糕的用户体验,若用户选取到不理想的应用,就有可能停止该动作序列,甚至卸载该应用,因此为用户推荐一个理想的移动应用执行序列很有必要。若能做到移动应用序列的个性化推荐,那么同样的功能在不同用户的手机上会调用不同的但更适合自己的移动应用序列来完成。

我们提出一个移动应用序列偏好度预测算法,用于解决应用执行序列的个性化推荐问题。利用序列偏好度预测算法计算用户对每个候选应用序列的偏好度,选取偏好度最大的应用序列作为推荐结果。

### 2.3.1 用户相似度矩阵

移动应用领域的推荐算法多采用协同过滤推荐(Collaborative Filtering Recommendation),协同过滤推荐利用已有用户群过去的行为预测当前用户最可能的行为<sup>[12]</sup>。我们采用John<sup>[13]</sup>提出的算法计算用户的兴趣相似度:

$$w_{uv} = \frac{\sum_{i \in N(u) \cap N(v)} \frac{1}{\log(1 + |N(i)|)}}{\sqrt{|N(u)| |N(v)|}} \quad (1)$$

$N(u)$ 、 $N(v)$ 分别表示用户  $u$  和  $v$  安装的应用集合,  $N(i)$  表示安装了应用  $i$  的用户集合, 该算法惩罚了用户  $u$  和用户  $v$  的共有应用集合中热门应用对相似度的影响。

### 2.3.2 推荐算法

假设某个开发者希望将这样一个功能集成进来: 通过拍照类应用或文件浏览类应用获取一幅图片, 接着通过图片编辑类应用进行图片处理, 最后通过社交类应用进行分享, 这样一个动作序列可以表示为:  $\langle A \cup B, C, D \rangle$ 。其中,  $A$  表示拍照类型动作,  $B$  表示文件浏览类型动作,  $C$  表示图片编辑类型动作,  $D$  表示分享类型动作。用户通过执行形如  $\langle a_1, c_1, d_1 \rangle$ 、 $\langle b_1, c_2, d_2 \rangle$  的应用序列完成该动作序列。现在的问题是如何利用这些应用序列构成的集合, 为没有使用该功能的用户  $u$  推荐最理想的应用序列。对于动作序列中的每一步, 用户  $u$  都有一个候选移动应用集合, 从每步的候选移动应用集合中选取一个应用, 即构成了一个候选应用执行序列  $\alpha$ 。从数据库中读取与用户  $u$  最相似的  $K$  个用户在完成该动作时采用的应用序列, 构成集合  $S$ 。

Sequence-Preference-Evaluation 算法借鉴了统计语言模型中的二元模型思想<sup>[14-16]</sup>, 利用集合  $S$  计算用户  $u$  对  $\alpha$  的偏好度。选取偏好度最大的候选应用序列, 即是用户的最优应用序列。

**算法 1** 计算用户对候选应用序列的偏好度 (Sequence-Preference-Evaluation 算法)

输入: 用户  $u$  的一个候选应用序列  $\alpha$ ,  $K$  个最相似用户的应用序列构成的集合  $S$ ;

输出: 用户  $u$  对候选应用序列  $\alpha$  的偏好度;

// 计算  $\alpha_0$  出现的频率

1. count := 0;

2. FOR  $\beta$  IN  $S$

3. IF ( $\beta_0 == \alpha_0$ ) count++;

4. pref := count /  $S$ .size;

// 循环计算  $\alpha_i$  出现的频率

5.  $i := 1$

6. WHILE  $i < \alpha$ .size

7. IF (pref == 0) RETURN 0;

8. numerator := 0, denominator := 0;

9. FOR  $\beta$  IN  $S$

10. IF ( $\beta_{i-1} == \alpha_{i-1}$ ) denominator++;

11. IF ( $\beta_{i-1} == \alpha_{i-1}$  & &  $\beta_i == \alpha_i$ ) numerator++;

12. pref \* = numerator/denominator;

13.  $i++$ ;

14. RETURN pref;

## 3 移动应用个性化集成框架实现

由于 Android 平台应用编程接口开放度较高, 有良好的运行机制支撑应用间相互调用, 因此基于 Android 平台初步实现了移动应用个性化集成框架。框架包含客户端和服务端两部分, 客户端为开发者提供了一套集成开发工具, 服务器端用于收集用户信息并将生成推荐的动作序列返回至客户端。

为初步展示实验效果, 我们在 SAE(Sina App Engine)上

搭建了一个简易服务器, 事实上服务器端也可以部署在其他平台上, 只需开发者修改相应的客户端配置文件, 在该框架下开发者无需对服务器端进行配置。下面详细阐述客户端和服务端端的实现。

### 3.1 客户端实现

客户端部分由 JAVA 语言编写, 以包的形式提供给开发者。客户端的整体架构如图 2 所示, 主要功能包括: 向服务器端上传用户信息; 从服务器端获取移动应用类别信息存储至本地数据库; 解析开发者编写的 ipel 脚本, 生成候选应用集合上传至服务器; 获取推荐的应用序列。

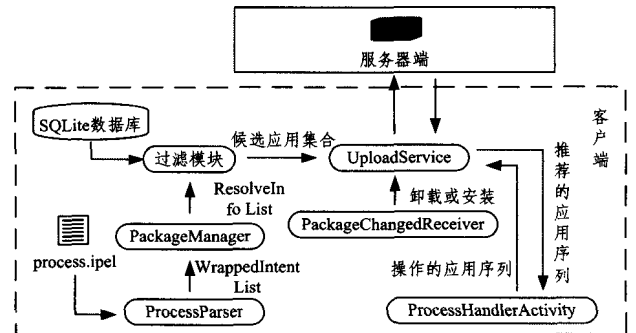


图 2 客户端整体架构

若一款应用基于本框架实现, 则用户在初次安装该款应用时, 本机的已安装应用信息会被上传至服务器, 这是由 UploadService 类完成的。该类继承于 IntentService 类, 负责完成客户端和服务端端的网络通信, 其他需要上传的数据包括: 动作序列中响应每个动作的经过筛选的候选移动应用集合、用户在完成某个动作序列所采用的应用序列。同时, 客户端 PackageChangedReceiver 类负责监听用户卸载旧应用或安装新应用类型的操作, 监听到这类操作后就通过 UploadService 类和服务器进行通信, 刷新用户的已安装应用信息。

客户端的另一重要功能是解析开发者编写的 ipel 脚本, 最终生成候选移动应用序列。用户编写的 ipel 脚本置于 Android 项目的资源文件夹 assets 目录下。ProcessParser 类基于开源 XML 解析工具 dom4j, 通过分析脚本中的  $\langle pick \rangle$  元素和  $\langle invoke \rangle$  元素, 生成自定义的 WrappedIntent 类, 按顺序放入表(List)中。Android 应用框架层提供了 PackageManager 类, 该类包含的方法 queryIntentActivities 可以分析 Intent 请求, 返回由 ResolveInfo 类型的对象构成的 List, 每个对象包含一个可以响应指定动作的具体移动应用信息。

这种基于 Intent Filter 机制得到的候选应用集合粒度过大。例如, 某开发者希望利用社交类应用分享一幅图片, 该动作对应的 Intent 请求中 action 属性值为 SEND, 解析该 Intent 请求后得到的候选应用列表中不仅包含微博、人人等社交类应用, 还包括网盘、邮箱、蓝牙等无关应用, 因此需要对应用集合进行过滤。考虑到谷歌市场(Google Play)将移动应用分为个性化、交通、体育等 34 个分类, 我们将分类概念引入到 IPEL 语法中, 每个  $\langle intent \rangle$  元素均包含这样一个子元素:

```
<extra key="category_in_google_play" value="No. ; No. ; ..."/>
```

该  $\langle extra \rangle$  元素的 value 属性由数字字符串序列组成, 每个数字字符串代表一个分类, 由分号隔开。例如微博和 Facebook 属于“社交类”应用, 编码为 25, 而网盘属于“效率类”应用, 编码为 23, 邮箱属于“工具类”应用, 编码为 27。利用编码

信息即可把响应同一 Intent 请求的不同类别的应用区分开来。

从数据库中读取这些应用所属的类别,筛选类型不符的应用,即得到了一个动作的候选应用集合,这就是过滤模块的功能。对每个 WrappedIntent 类均进行此操作,即可得到该动作序列中每步的候选应用集合,并将其上传至服务器。

Activity 作为 Android 最重要的组件之一,是应用形成的基础,负责系统和用户间的交互。但是它在响应一次应用请求后即返回,为克服这种“一步式”响应机制,我们自定义了 ProcessHandlerActivity 类。该类继承于 Activity 类,重写了 onActivityResult 方法,在响应了一次应用请求后,检测是否有后续请求,若有,则发起新的 Intent 请求,以此实现“多步式”顺序响应机制,达到“序列化复用”已安装应用的目的。在得到推荐的移动应用序列之后,该类在运行时刻绑定已安装应用和用户进行交互。为了展示响应 Intent 请求的候选应用集合,并且记录候选集合的点击事件,需要自定义一个 Intent 请求响应界面,这部分实现也包含在该类中。

### 3.2 服务器端实现

服务器端由 PHP 语言编写,整体架构如图 3 所示,主要功能有:向 Google 服务器提交 HTTP 请求,获取某个移动应用的类别信息并返回至客户端;根据用户已安装的应用信息计算用户相似度矩阵;为用户推荐理想的应用序列。

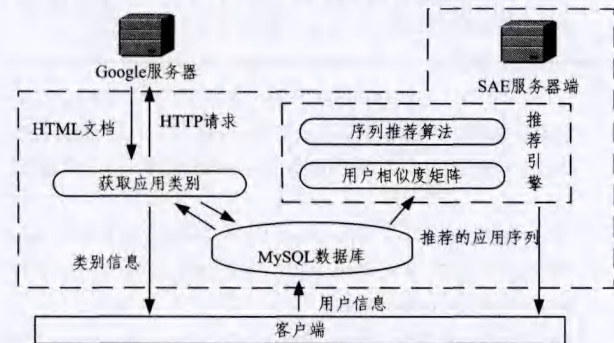


图 3 服务器端整体架构

用户在第一次使用基于本框架开发的移动应用时,手机中已安装的应用列表会上传至服务器数据库。获取移动应用类别模块向 Google 服务器发送携带应用包名信息的 HTTP 请求,返回包含指定应用的页面信息。PHP 语言自定义的 DOMDocument 类可以解析 HTML 文档,从中获取类别信息,返回至手机客户端。

服务器端的数据库中存储的信息是“用户-应用”信息对,若对两两用户直接使用式(1)计算相似度,则时间复杂度过高。为此,首先建立应用到用户的倒排表<sup>[17]</sup>,接着利用式(1)计算用户间的相似度,得到用户相似度矩阵。

客户端会将动作序列的候选应用集合上传至服务器,从每个动作的候选应用集合中任意选取一个应用,即组成了一个候选应用序列  $\alpha$ ;从用户相似度矩阵中选取  $K$  个最相似的用户,从数据库中读取他们关于此动作序列的应用序列,构成应用序列集合  $S$ 。 $\alpha$  和  $S$  即构成了 Sequence-Preference-Evaluation 算法的输入,输出即是用户对于该候选应用序列的偏好度。从所有可能的应用序列组合中,选取偏好度最大的应用序列,即可得到推荐结果。

### 3.3 集成开发工具使用实例

开发者在使用该工具时,需要在 Android 项目中导入我们编写的 JAVA 包,另外需要在 AndroidManifest.xml 中对

UploadService 和 PackageChangedReceiver 进行注册,同时添加读取手机状态和连接网络的权限。

下面模拟一个开发场景以说明开发者如何使用该工具。假设某开发者希望在应用中集成这样一个动作序列:首先通过拍照或文件浏览类型的动作获取一幅图片,接着通过图片编辑类型的动作对图片进行处理,最后通过分享类型的动作将图片分享至社交网络。开发者首先遵循 IPEL 语法规则编写 image\_share.ipel 脚本文件。篇幅所限,这里仅举例图片编辑类型的动作在脚本中如何定义:

```
<request describe="edit the pic" skip="false" reselect="false">
  <invoke>
    <intent>
      <action
        actionname="android.intent.action.SEND"/>
      <data type="image/jpeg"/>
      <extra
        key="android.intent.extra.STREAM"
        value="file:///mnt/sdcard/test.jpg"/>
      <extra
        key="category_in_google_play"
        value="22;"/>
    </intent>
  </invoke>
</request>
```

开发者将包含该功能的 Activity 组件的父类置为 ProcessHandlerActivity 类,接着在启动该动作序列的代码段调用 startProcess 方法即可触发应用序列。以开发者在按下一个 Button 时执行该动作序列为例,执行过程如图 4 所示。

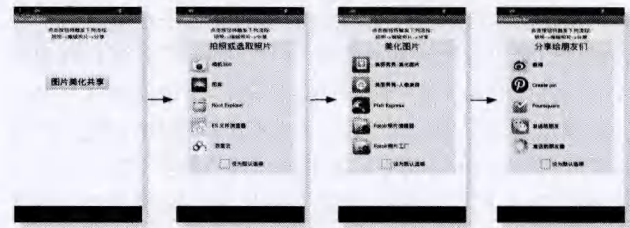


图 4 一个动作序列的执行过程

如图 4 所示,开发者所定义的 ipel 脚本被解析为一个完整的动作序列。在按下 Button 后触发该动作序列,每个动作都有一个对应的经过筛选的候选应用集合。若与服务器端进行通信则会获取一个推荐的应用序列,如“ES 文件浏览器—> Fotor 照片编辑器—> 微博”,此时图 4 中的候选应用列表不会弹出,而是直接进入指定应用,这样可以节省用户时间。可以看出,不同用户在使用该功能时会基于本机已安装应用列表得到不同的应用序列,这种机制为用户带来了更好的使用体验。

**结束语** 本文提出了一个移动应用个性化集成框架,基于该框架,开发者可以从更为自然的执行流程角度对已有应用进行序列化复用,从而便捷地构造新应用。同时,该框架基于应用序列偏好度预测算法,为用户推荐理想的应用执行序列以期提升用户体验。

该框架还存在不足和有待提高之处,本文初步展示了该框架的可行性,没有考虑更多的细节问题。在将来的工作中,可以进一步完善该框架,降低集成开发工具的使用难度;在动作序列推荐部分,可以考虑增加上下文变量等更多信息,更加

## 参考文献

- [1] <http://developer.android.com/guide/components/intents-filters.html>
- [2] 孟祥武,胡勋,王立才,等. 移动推荐系统及其应用[J]. 软件学报,2013,24(1):91-108
- [3] Woerndl W, Schueller C, Wojtech R. A hybrid recommender system for context-aware recommendations of mobile applications [C]// 2007 IEEE 23rd International Conference on Data Engineering Workshop. IEEE, 2007: 871-878
- [4] Yan Z, Zhang P, Deng R H. TruBeRepec: a trust-behavior-based reputation and recommender system for mobile applications[J]. Personal and Ubiquitous Computing, 2012, 16(5): 485-506
- [5] Shi K, Ali K. GetJar mobile application recommendations with very sparse datasets[C]// Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2012: 204-212
- [6] Yan B, Chen G. AppJoy: personalized mobile application discovery[C]// Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. ACM, 2011: 113-126
- [7] Davidsson C, Moritz S. Utilizing implicit feedback and context to recommend mobile applications from first use[C]// Proceedings

- of the 2011 Workshop on Context-awareness in Retrieval and Recommendation. ACM, 2011: 19-22
- [8] Bray T, Paoli J, Sperberg-McQueen C M, et al. Extensible markup language (XML)[J]. World Web Journal, 1997, 2(4): 27-66
- [9] Fallside D C. XML Schema Part 0: Primer [EB/OL]. <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>, 2001
- [10] Thompson H S. XML Schema Part 1: Structures [EB/OL]. <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>, 2001
- [11] Biron P V. XML Schema Part 2: Datatypes [EB/OL]. <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>, 2001
- [12] Jannach D, Zanker M, Felfernig A, et al. Recommender systems: an introduction. Cambridge University Press, 2010
- [13] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]// Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., 1998: 43-52
- [14] 吴军. 数学之美[M]. 北京: 人民邮电出版社, 2012
- [15] Brown P F, Cocke J, Pietra S A D, et al. A statistical approach to machine translation[J]. Computational linguistics, 1990, 16(2): 79-85
- [16] Rabiner L, Juang B. An introduction to hidden Markov models [J]. ASSP Magazine, IEEE, 1986, 3(1): 4-16
- [17] 项亮. 推荐系统实践[M]. 北京: 人民邮电出版社, 2012

(上接第 49 页)

表明,本文提出的测试方法确实能够模拟出真实用户对 Web 应用造成的负载,证实了本方法的有效性。

相比于传统的录制回放测试方法以及其他的基于用户会话的测试方法,本文提出的方法具有以下 3 个优点:(1)该方法能够自动生成性能测试用例,省去录制和设计测试脚本的工作;(2)借助真实用户的用户会话信息,可以使测试脚本产生的模拟负载与真实负载相一致,保证了性能测试过程的真实性和测试结果的有效性;(3)相比于其他的基于用户会话的测试方法,本文方法使用了生物序列比对算法,将用户会话中的 URL 请求看作是一组序列而不是一组集合,保障了用户会话的序列性。

未来的研究工作将会集中在以下几个方面:(1)增强对无效日志文件的识别;(2)对用户会话相似度算法进行优化改进,降低算法和空间复杂度;(3)挖掘更多的日志信息,例如针对移动互联网应用,日志信息提供的浏览器信息可以把不同品牌的手机用户进行区分(不同的浏览器对移动互联应用发出的请求,服务器端返回的数据将有所不同),产生不同的测试用例;(4)研究基于用户会话的性能测试方法的性能评估。

## 参考文献

- [1] Elbaum S, Rothermel G, Karre S, et al. Leveraging User-Session Data to Support Web Application Testing[J]. IEEE Transactions on Software Engineering, 2005, 31(3): 187-202
- [2] Myers E, Miller W. Optimal alignments in linear space [M]. Computer Applications in the Biosciences (CABIOS). 1988: 11-17
- [3] Rudolf A, Pirker R. E-Business Testing: User Perceptions and Performance Issues[C]// Proceedings of the First Asia-Pacific Conference on Quality Software. IEEE Press, 2000: 231-239

- [4] Subraya B M, Subrahmanya S V. Object Driven Performance Testing of Web Applications[C]// Proceedings of the First Asia-Pacific Conference on Quality Software. IEEE Press, 2000: 117-125
- [5] Mercury Interactive Corporation. Load Testing to Predict Web Performance[R]. Technical Report WP-1079-0604. Mercury Interactive Corporation, 2004
- [6] Liu Yue, Wang Kang, Wang Wei, et al. User-session-based Test Cases Optimization Method based on Agglutinate Hierarchy Clustering[C]// IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing. 2011: 414-418
- [7] Quan Xiu-xia, Lu Lu. Session-Based User Behavior Meta-Model of Web Applications for User-Level QoS Load Testing[C]// Supply Chain Management and Information Systems (SCMIS). 2010: 152-167
- [8] Wang Wei-nan, Zaiane O R. Clustering Web Sessions by Sequence Alignment[C]// Proceedings, 13th International Workshop on Database and Expert Systems Applications, 2002. 2002: 394-398
- [9] Needleman S, Wunsch S. A general method applicable to the search of similarities in the amino acid sequence of two proteins [J]. Journal of Molecular Biology, 1970: 443-453
- [10] Adrian D, Pau L, Jonathan S, et al. FastLSA: A Fast, Linear-Space, Parallel and Sequential Algorithm for Sequence Alignment[C]// Proceedings of the 2003 International Conference on Parallel Processing. 2003: 304-309
- [11] Diwakar K, Jerome R, Shikharesh M. A Synthetic Workload Generation Technique for Stress Testing Session-Based Systems [J]. IEEE Transactions on Software Engineering, 2006, 32(11): 868-882