

Android 多窗口系统的研究、设计与实现

罗浩^{1,2} 王丹¹ 肖络元² 陈渝²

(北京工业大学计算机学院 北京 100124)¹ (清华大学计算机科学与技术系 北京 100084)²

摘要 多窗口技术作为一种改善人机接口的计算机技术,已在桌面系统中流行。Android 系统作为手机系统具有广泛的用户群体,并逐渐扩展到大屏幕。但当前的 Android 系统缺少多窗口的有效支持,这极大地限制了使用者在大屏幕环境下使用 Android 系统。在 Android 的基础上,着眼于桌面平台,通过分析目前 Android 多窗口系统的功能需求,设计并实现了一套 Android 多窗口系统;通过对该系统的显示效果以及性能指标进行评估,阐述了 Android 多窗口系统的可用性和实际意义。

关键词 Android, Android 窗口管理, 桌面化操作系统

中图分类号 TP316 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.11.019

Research, Design and Implementation of Android Multi-window System

LUO Hao^{1,2} WANG Dan¹ XIAO Luo-yuan² CHEN Yu²

(School of Computer Science, Beijing University of Technology, Beijing 100124, China)¹

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)²

Abstract Multi-window technology which is regarded as a way to improve the man-machine interface, has been popular in design of desktop system. As a phone system, Android has a large amount of user groups, and gradually expands to the large-screen devices. However, Current Android systems lack supports from multi-window, which greatly restricts users to use the Android system in the large-screen environment. Based on Android, focusing on the desktop platform, a multi-window system was designed and implemented for Android through the analysis for the functional requirements of the Android multi-window system. The availability and practical significance of Android multi-window system was assessed by displaying the effects and performance of the system.

Keywords Android, Android window manager, Desktop OS

1 引言

Android 是当前使用范围最广的移动端操作系统,近些年,用户对 Android 的应用需求正在发生转变。随着大屏幕设备的逐渐增多以及设备性能的不提高,传统上用整个屏幕显示一个内容的显示模式已经逐渐开始显得乏力。通过对众多用户需求的分析,Android 桌面化已经是当前 Android 的一个重要的发展方向。本文选择桌面化操作模式的最关键点——多窗口作为切入点来进行研究,将 Android 运行在性能高、屏幕大的桌面计算机上。本文将首先对 Android 的窗口系统以及国内外的一些相关研究进行调研总结,并通过分析其共性来寻找实现多窗口的关键点;然后在全面分析多窗口机制的基础上,分别从整体 UI 系统和 APP, Activity, Window 子系统的角度进行设计与实现。最后,在物理计算机上进行实际测试与评估,并进行相关总结。

2 相关工作

2.1 Android 系统相关项目的介绍

Android 是一个开源的、用于移动设备的系统。为了促进 Android 的发展,谷歌领导的开放手机联盟(Open Handset Alliance)的一系列公司创立了将 Android 代码进行开源的 AOSP(Android Open-Source Project)计划,以推广 Android 系统,同时他们号召各界开发人士来对 Android 系统进行进一步完善。通过 AOSP 能够从官方渠道获取最纯正的 Android 系统源代码,并在此基础上进行开发,这种资源共享方式极大地促进了 Android 系统的发展。同时对于开发者而言,AOSP 为他们提供了一种更加便捷的方式来开展 Android 系统的研发,这同时也扩充了 Android 系统的市场。

Android-x86 项目^[2]是以 AOSP 项目为基础进行的系统定制项目。该项目的目的在于将 Android 系统在大部分 x86

到稿日期:2016-10-16 返修日期:2016-12-27

罗浩(1991—),男,硕士生,主要研究方向为 Android 系统 UI、linux 内核, E-mail: ladehunter@gmail.com(通信作者);王丹(1969—),女,博士,教授,博士生导师,主要研究方向为可信计算、数据流管理;肖络元 男,硕士生,主要研究方向为 linux 内核;陈渝 副教授,硕士生导师,主要研究方向为操作系统及以操作系统为核心介入的普适计算等。

平台上进行适配。利用 x86 架构设备的高性能来呈现一个与以往不一样的 Android 操作体验。该项目由 Cwhuang 和 Be-younn 主持设计。

考虑到桌面环境更适用于本文所实现的多窗口系统的测试评估,因此本文在基于 AOSP 的同时也参考了 Android-x86 项目的内容,更好地测试和评估多窗口系统的实现效果。

2.2 国内外相关研究

多窗口界面旨在显示并完成与用户之间的交互,为此,在具体设计上,其往往是数据结构、处理过程、交互、控制等多种因素共同作用的结果。多窗口界面的典型特征是:每个窗口在系统管理和界面管理的部分均通过自身独立的消息系统来完成。同时,系统也提供了统一的管理系统来处理窗口内部消息的反馈结果和窗口间消息的互相影响,形成了从系统控制到应用控制的多层次结构化管理模式。

Jabol^[9]多窗口系统是 Tieto 公司实现的基于 Android4.4 的多窗口系统,其实现思路能够体现 Google 的设计理念和 Android 的内在特点。但是其对窗口结构的设计不够严谨,对 Android 窗口样式的考虑不够全面,导致其在应用兼容性和窗口样式适配方面存在不少的问题。例如其多窗口边框作为 Android 原有窗口根的子 View 加入,导致很多应用程序能够对其进行操作,使其失去功能,从而造成应用程序不兼容等问题。

相比于 Jabol,remixOS^[10]和 PhoenixOS^[11]都是实现程度已经相当高的桌面化 Android 系统。两者均通过对 Android 原有窗口结构的大规模推倒重建并构建类似 Windows 的系统应用功能来实现相对成熟、兼容且易用的操作系统。但其缺点在于设计思路与 Android 自身功能及理念偏差较大,实现过于复杂,随着 Android 系统版本的不断提升,其稳定性难以得到保证。

针对上述不足,本文对当前 Android 的窗口系统进行重新设计,使其能够在大屏幕下有效地实现多窗口功能,且在 CPU 性能和内存占用量上与现有的单窗口模式的 Android 系统持平。

3 系统设计与实现

3.1 Android 多窗口的整体架构

Android 多窗口管理系统主要通过 3 个大的功能子系统共同合作来完成工作,即 APP 子系统、AMS (ActivityManager-Service) 子系统和 WMS (WindowManagerService) 子系统。APP 子系统负责一个窗口本身所要接受的消息事件以及 Window 和 View 之间关系的适配;AMS 子系统负责管理和服务系统中所有的 Activity 集合;WMS 子系统负责管理管理系统中的所有窗口之间的关系。其总体架构如图 1 所示。下面分别从这 3 个子系统各自的功能以及子系统联系的方面来全面介绍 Android 的窗口管理系统。

APP 子系统是用户直接与窗口管理系统交互的接口,也是整个窗口管理结构中唯一能够直观操作的部分。因此 App 部分与 AMS 和 WMS 都存在一定的联系。从用户的角度来看,APP 子系统是整个窗口管理系统的核心。设计 App 功能时主要分为用户 App 和系统 App 两部分。在用户 App 部分

需要实现包含窗口化基本操作按钮的 Header 和用以直观反映窗口边界并具备缩放功能的 Border,以及充分适配窗口样式和应用操作,以保证两者正确地与应用本身的显示内容相结合。在系统 App 部分则需要实现类似 Windows 开始菜单、任务栏等的功能,以便用户更自由地进行窗口管理操作,同时搭建 APP 子系统与 AMS, WMS 两个控制子系统的通路。

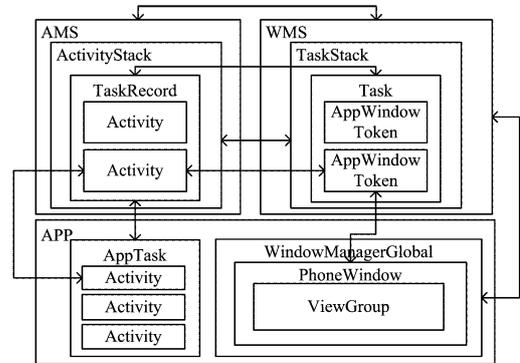


图 1 多窗口系统的总体架构图

在 AMS 中,一个 Activity 会通过一种层级记录的方式进行管理。其中 ActivityStack 被用来控制其中每一个 Activity 所作用的屏幕范围。有了屏幕和作用范围之后,由 TaskRecord 进行集中管理。多窗口系统中 ActivityStack 的新建及其 Focus 情况更改是 AMS 需要实现的主要功能。每个新启动的应用都应该有其独占的 ActivityStack 来保证应用能够获得独占的屏幕空间。在 Focus 变更时需要 AMS 正确处理各种类型应用之间的前后顺序关系以帮助 WMS 计算符合逻辑的显示状态。

WMS 的管理方式主要通过应用的行为来决定如何对窗口进行管理,通过应用输入和事件反馈对窗口的显示叠加进行计算,并将显示内容发送给底层显示子系统来完成窗口的显示。本系统中 WMS 核心功能是制定窗口的叠加策略及设置窗口的屏幕区域。Android 中有多种窗口样式,不同样式窗口的叠加需要策略进行管理才能够正确进行。同时,为了辅助 AMS 中 ActivityStack 的屏幕区域划分,WM 需要为窗口进行屏幕区域的分配并且使之与 Activity 屏幕限定区域保持一致,以确保应用交互范围的正确性。

3.2 Android 多窗口系统的工作流程

Android 多窗口系统的工作流程主要分为窗口生成和窗口刷新。

3.2.1 窗口生成

Android 的窗口是通过应用的启动来生成的,而应用的启动则是依靠 Activity 的启动进行生成的。本文所设计的多窗口系统采用如图 2 所示的启动流程。对于启动流程而言,本文所做的工作主要集中在图 2 中创建 ActivityStack 和构造 PhoneWindow 两个部分。ActivityStack 能够限定其中所有 Activity 的活动屏幕区域,那么从多窗口的设计角度来讲,通过建立多窗口来摸索 ActivityStack 的新建策略,任一新启动应用的 Main Activity 将开启一个新的 ActivityStack,以此保证新的应用能够直观地出现在一个新的窗口上。为支持多窗口模式的窗口外层显示,需要实现窗口化操作的诸多辅助功能,因此在构造 PhoneWindow 时需要同时构造承载这些辅助

功能的 Header 和 Border 两个控件。在实现上述功能后,就能生成一个多窗口模式下的窗口。



图2 窗口生成流程图

3.2.2 窗口刷新

多窗口模式下的 Android 窗口刷新流程图如图 3 所示。这部分的实现对象涉及 AMS 和 WMS 两部分。对 WMS 而言,最关键的是要重新限制应用的显示范围,同时完成应用窗口的显示叠加以及前后层次顺序的判断,以保证多个应用在多窗口模式下的正确运行。

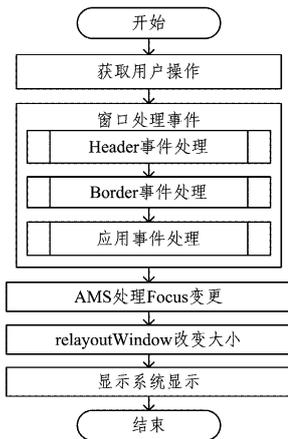


图3 窗口刷新流程图

3.3 APP 子系统

PhoneWindow 是 APP 子系统中主要关注窗口绘制和功能模块,也是多窗口模式下 APP 子系统对窗口的实际作用对象。在多窗口系统下需要 APP 子系统的 PhoneWindow 功能模块与 AMS 和 WMS 进行协同操作。APP 子系统的实现主要分为系统 App 和应用 App 两部分。

为了实现桌面化操作模式,系统 App 与应用 App 需要相互协调和相互操作。通过对 Linux/Windows 桌面多窗口系统的分析,实现了由开始菜单和任务栏组成的多窗口模式 SystemUI。其中开始菜单主要负责实现加载启动应用的功能,在开始菜单中实现加载启动应用功能可以给用户提供一种更加便捷的应用启动方式,其实现主要采用获取应用列表方式和点击启动方式。为了使用户操作窗口具有便携性,本系统对任务栏进行重新设计,通过在任务栏上对已启动的应用添加对应的多模态图标显示,并对对应应用 App 的最小化

功能实现窗口回复功能来进一步优化整体的桌面化操作体验。

应用 App 相关功能的实现分为两部分:辅助功能的实现和窗口样式的适配。

在辅助功能方面,主要实现外框顶部的 Header 和外框四周的 Border。Header 主要负责窗口移动和 4 个按钮:关闭、最大化-恢复、最小化以及后退。窗口大小变化相关部分的功能主要通过设置 TouchListener 并调用前面提到过的relayoutWindow方法来进行实现,以保证实现方式的安全可靠。而后退、关闭等功能则同样是通过关联之前实现在系统界面上的后退以及关闭的事件响应来实现。我们实现了类似主流桌面操作系统的 Border 缩放方式。这里的缩放功能同样也是通过类似于最大化最小化的方式实现,因此不再赘述。

样式适配功能的主要目的是保证窗口结构的正确性以及 Header 和 Border 的正确添加,让不同的窗口样式都能够显示出 Header 和 Border 以便多窗口的使用及操作。为了适配应用对 DecorView 的操作,需要保证新加入的 Header 和 Border 对应用来说是不可见的,因此将 Header 和 Border 在 PhoneWindow 中的层次放在 DecorView 上面,保证 Header 和 Border 相比于 DecorView 处于更高一级并保证其相对于应用而言有着更高的优先级以及对应用的不可见性。由于 Android 在窗口构造过程中调用作为资源文件的窗口样式 xml 文件并读取其中记录的窗口样式信息来完成窗口构造,为了与之适配,需要在多窗口模式下适配单窗口的特殊样式,以此来保证 Header 和 Border 的应用范畴以及窗口样式适配的正确显示。

相对于 AMS 和 WMS 而言,APP 子系统的功能细节较多,需要设计的要点也更多,其主要功能在于实现多窗口操作模式的必要功能以及从 Window 和 View 联系的角度来兼容窗口样式和应用的运行。

3.4 AMS 子系统

如上一节所说,AMS 方面的功能实现要点是围绕 ActivityStack来进行功能的实现与扩展。其中最主要的功能在于两个方面:

(1)ActivityStack 的生成机制。对于多窗口而言,每一个 ActivityStack 对应着一块屏幕区域,用来与用户交互。若应用不能够获得独立的 ActivityStack,则会出现多个应用共用一块窗口区域的情形。因此需要设计多窗口模式的 ActivityStack生成机制,其核心思想是通过判断当前 Activity来源,让应用的 Main Activity 能够自行启动一个 ActivityStack;而通过其他应用的 Activity 来启动的应用则进行应用间状态的判断,并让应用程序可以得到一块专门的屏幕区域来显示交互内容。

(2)应用的焦点的变化。Android 设计了两类 ActivityStack,一类是一般应用使用的 AppStack,另一类是 Home 系统应用(即通常所说的 Launcher,相当于桌面显示)所使用的 HomeStack。为了保证 Home 系统应用不会覆盖应用窗口,多窗口模式的 Focus 机制采取将 Home 系统应用一直保持在最下方的设计方式。重获 Focus 的 APPStack 会直接显示到

最上方,其余 APPStack 则按照相应的顺序排列,这样就保证了 Home 系统应用不会覆盖应用,且一般应用自身也通过 AMS 原有的排序功能维持正确的前后层叠关系。对于 Android 原生系统而言,处于非 Focus 状态的应用被系统认定为不可见,因此会进入暂停状态,而对于多窗口而言,没有最小化的应用都应该让其后台运行。我们通过对 Android 的应用暂停机制的重新设计,在使应用进入暂停状态时对其所处屏幕位置进行判断,只有当应用在屏幕外部时才将其暂停。这种做法一方面能够让可见的应用继续保持运行(但不能够响应操作),另一方面能够让应用程序在设备共享方面有一个自然的策略。由于原本通过 Activity 的方式以用户操作为触发进行设备交互的应用程序得不到用户操作的响应,因此也不会触发其余共享设备的数据交互。而原本通过 Service 方式来使用设备的应用程序则可以继续在后台通过与设备的数据互通来更新应用程序的状态。这合理地解决了多窗口应用间共享设备的问题。

3.5 WMS 子系统

WMS 的主要功能是处理窗口之间的关系以及窗口变化对其他窗口的影响。这需要关注不同的窗口样式、窗口结构以及窗口之间的关系。

对多窗口而言,WMS 也有两方面的内容需要进行重点实现:

第一方面是对出窗口区域的限制。它需要通过 WMS 中与 ActivityStack 对应的 TaskStack 类的 `relayoutWindow` 方法来实现。这样能够保证该方法通过 TaskStack 与 ActivityStack 进行交互,进而保证 Activity 和 Windows 的屏幕范围一致。

另一方面是窗口叠加策略。Android 为不同的窗口样式设计了层次关系,各个窗口样式之间有着绝对的叠加重关系,为了保证窗口的叠加情况正确,实现了一套桌面化的窗口叠加重,对系统 APP 的窗口给予更高的权重(例如系统警告、开始菜单、任务栏等)使其能够显示在所有的应用窗口之上。而对于一般应用 APP 的窗口,则统一权重,并按照 Focus 情况进行排列。

WMS 的功能设计和实现更多的是在多窗口的情况下保证窗口之间影响的适配,使得多个窗口的同时显示能够尽可能正确与完整。由于窗口样式是 Android 中一个比较复杂的部分,也是 Android 对应用程序多样化的一种基本支持,因此

WMS 功能的设计思路虽然相对简单,但在实现上则存在着较大的细节问题需要处理。

4 测试与评估

采用主流桌面计算机系统(CPU: I7-2670QM-2. 2GHZ, 内存:4GB)进行多窗口 Android 系统和原生 Android 系统的分析对比测试,分别甄选了系统应用和第三方应用作为测试用例,系统应用包括浏览器、计算器、下载管理器。第三方应用是腾讯 QQ。根据应用的运行情况对其兼容性进行评估。

4.1 实现效果展示

通过对多窗口系统的实现,本系统目前能够达到的效果如图 4 所示。可以看到,多窗口运行模式以及大多数应用能够正常完成和显示,失去焦点的应用也能够继续执行。少部分应用由于窗口样式的适配问题出现了显示的问题,这符合上文提到的情况,Android 中的窗口样式众多,在今后的改进中适配更多的样式将是实现的重点。



图 4 窗口叠加显示效果示意图

4.2 应用兼容性测评

本节将通过对主流应用运行在多窗口系统下的适配情况进行分析来测评应用在多窗口 Android 系统上运行的兼容性。

由于本系统相对原生 Android 系统而言的最大改进主要在于 Framework 部分,因此应用的兼容性问题主要是因窗口样式的适配导致的。表 1 分别对几类窗口的适配问题和解决方案进行了总结。从表 1 中可以看出,对于大部分窗口样式问题,其都能够通过将原生 Android 使用的样式和多窗口系统使用的样式通过分开处理的方式来进行解决,对于少量应用自带样式与多窗口系统不适配的问题,则需要对应用进行针对处理。同时,也在其他多窗口系统上进行了应用兼容性测试,结果与本系统的结果基本类似,这也说明这类问题是 Android 系统在多窗口实现上必须要解决的问题。

表 1 窗口样式适配问题与解决方案

窗口样式	应用名称	问题表现	解决方案
固定大小窗口适配问题	下载管理器、WPS 等	应用部分窗口大小固定,多窗口系统下无法完全显示或显示出错	针对该样式建立对应的多窗口样式,并匹配窗口大小
浮动窗口适配问题	部分 SystemUI、文件管理器、WPS、微信、QQ、豌豆荚等	浮动窗口样式带来的屏幕阴影和多窗口系统下的显示出错	对 SystemUI 中浮动窗口禁用多窗口模式,对应用则建立对应的多窗口样式并匹配窗口显示
SystemUI 干涉适配问题	绝大多数全屏显示的应用	打开部分应用会导致状态栏等 SystemUI 部分的显示出错	多窗口下禁止应用对 SystemUI 的显示模式进行修改
弹出窗口适配问题	WPS、笔记本、通讯软件	弹出窗口显示位置错误	针对该样式建立对应的多窗口样式,并建立弹出窗口的全窗口相对坐标到屏幕绝对坐标的映射关系
应用自带窗口样式适配	QQ、微信	部分窗口点击无响应	单独判断应用,进行针对处理

4.3 性能测评

本节将通过整体性能测试和应用性能测试两部分来对比多窗口 Android 系统与原生 Android 系统以及前文中提到的其他 Android 多窗口系统的性能,并以此来性能评估。在整体性能测试方面,采用性能测试应用 Antutu 的测试评分作为基准进行对比。数据获取方式为通过 Antutu 应用和系统命令。Antutu 评分对比如图 5 所示,内存占用量如图 6 所示,CPU 占用量如图 7 所示。根据 Antutu 的评分,可以认为本系统对 Android 的整体性能表现基本没有影响。应用程序的内存占用量均有小幅上升,我们认为这种情况应该是由 Header 和 Border 的添加而导致的,属于正常现象。各系统的 CPU 占用量基本相同,可以认为系统功能的实现没有增加应用的 CPU 占用。从横向来看,本系统和其他多窗口系统在性能方面没有明显差异。这说明本系统在采用更易于升级、更符合 Android 系统原本设计理念的设计方法和实现方法的基础上,在性能上与同类系统以及 Android 原生系统持平,并没有为了兼容性和易维护、易升级的目的而牺牲性能。

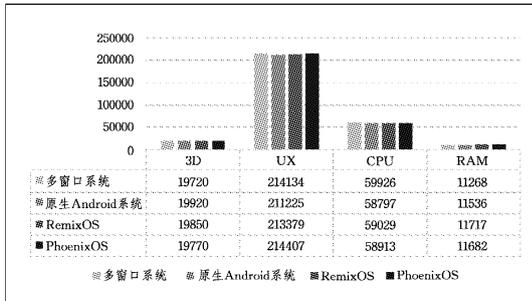


图 5 Antutu 性能评分统计

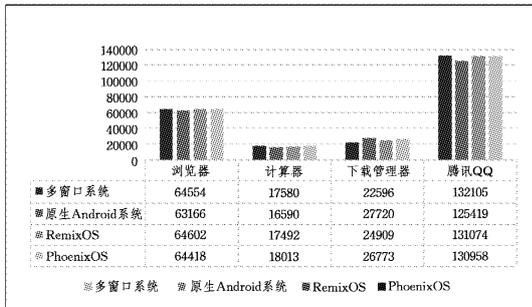


图 6 应用内存占用统计

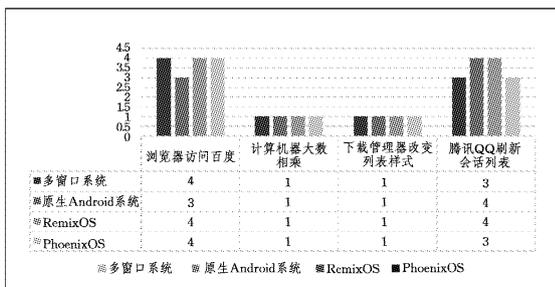


图 7 应用行为 CPU 占用量统计

结束语 本文以 Android 为目标平台,在考虑已有的 Android 桌面化研究的基础上对 Android 多窗口系统所需要的功能和所要实现的效果进行了研究和总结。之后通过聚焦 Android 固有的窗口管理系统,对其功能的设计思路、实现方

式以及工作模式做了深入的考察与分析,结合多窗口的功能需求,分别从 APP, AMS 和 WMS 3 个子系统对多窗口功能的切入点进行了分析和总结,并以尽可能使用 Android 自身提供的接口和方法的理念为指导,设计并实现了具备基本多窗口显示功能和窗口化操作功能的多窗口系统。在显示效果和性能指标两方面,以 App 的单独显示和 App 的叠加显示为测试主题进行了实现效果的评估。从结果来看,本系统已经具备了较为完善的多窗口显示机制,同时也能够提供基础的桌面化操作模式,具备初步的可用性,并且在 CPU 的占用和内存消耗上与原有 Android 相比没有增加。

本系统在显示效果方面仍存在不足,但从文中提到的对于个别应用的兼容性问题解决情况来看,这主要是因为 Android 系统中海量的窗口样式以及多种自定义 View 和原有 View 的组合所带来的问题。从对已发现的兼容性问题的分析以及解决方法的经验结果来看,这类问题都是可以解决的,这也从侧面证明了本系统的设计理念和实现方向没有问题。同时,在通过适配更多窗口样式而彻底改进应用兼容性的基础上,下一步的目标是提升桌面化用户体验。这方面有很多成功的个案值得我们学习。在兼容性和用户体验得到进一步提升之后,本文所实现的 Android 多窗口系统定能够成为一款可用、易用且好用的桌面化操作系统。

参考文献

- [1] Google-Android Open-Source Project[OL]. <http://source.android.com>.
- [2] Android-x86 Open Source Project[OL]. <http://www.android-x86.org>.
- [3] Blog. Android Window management analysis sdn[OL]. <http://net/huanxido/article/details/7879529>.
- [4] XIA D B, CHEN Q K. Activity Window Management System in the Android applications[J]. Computer Systems & Applications, 2012, 21(6): 224-227. (in Chinese)
夏德冰, 陈庆奎. Android 应用中一种 Activity 窗口管理系统[J]. 计算机系统应用, 2012, 21(6): 224-227.
- [5] LIU C P, LIU Y, CHEN J S. Study on Dynamic Interface Layout for Android Application[J]. Computer and Modernization, 2013 (5): 197-200. (in Chinese)
刘昌平, 刘洋, 陈佳实. Android 应用程序界面动态布局研究[J]. 计算机与现代化, 2013(5): 197-200.
- [6] foocoder.com android Window management framework analysis [OL]. http://www.cnblogs.com/noTice520/arc_hive/2012/10/25/2738050.html.
- [7] 邓平凡. 深入理解 Android: 卷 I[M]. 北京: 机械工业出版社, 2011.
- [8] 韩超, 梁泉. Android 系统原理及开发要点详解[M]. 北京: 电子工业出版社, 2010.
- [9] Tieto multiwindow for android[OL]. https://github.com/tieto/multiwindow_for_android.
- [10] Jide Co. remixOS[OL]. <http://cn.jide.com/remixos>.
- [11] ChaoZhuo Co. PhoenixOS[OL]. <http://www.phoenixos.com>.

网购平台搜索功能的质量评估方法研究

陈浩¹ 陶传奇¹ 文万志²

(南京理工大学计算机科学与工程学院 南京 210094)¹ (南通大学计算机科学与技术学院 南通 226019)²

摘要 随着大数据的迅速发展,大数据应用层出不穷,诸如网购零售平台、人脸识别系统、智能决策系统、自助客服、看病导医系统等典型的大数据应用使得人们的生活越发便捷。搜索系统是人们最常使用的大数据应用之一。然而,搜索系统在不同平台上的功能各有侧重,其标准尚且不完善,搜索的质量参差不齐,无法得到保障。与普通的文本搜索引擎相比,网购平台的搜索引擎增加了分类检索、筛选等特色功能,其质量的评价与保障更为复杂。通过对网络零售平台的搜索功能进行研究,针对网购平台搜索功能的质量评价提出了质量参考因素,针对质量因素提出了若干评价指标以及相应的实现算法,并通过实验来论证了质量指标的有效性。

关键词 搜索引擎,质量因素,质量指标,质量评估,网购搜索

中图分类号 TP311 文献标识码 A DOI 10.11896/j.issn.1002-137X.2017.11.020

Research on Quality Evaluation Approaches for Search Function of Online Shopping Platforms

CHEN Hao¹ TAO Chuan-qi¹ WEN Wan-zhi²

(School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)¹

(School of Computer Science and Technology, Nantong University, Nantong 226019, China)²

Abstract With the rapid development of big data, various big data applications are now in service in different fields. Typical big data applications, such as online shopping and retailing platform, face recognition system, intelligent decision system, self-help service system, medical treatment system make daily life more convenient. Search system is one of the most used big data applications. However, search system varies in different platforms, and there are few standards for it. The quality of search system is hard to assure and validate. Search engine for online shopping systems combines text search and classification-based retrieval comparing to common text search engines. It is harder to validate and evaluate quality of it. Through studying on search system of online shopping platform, some quality factors and relational implementation algorithm were provided in this paper to validate and evaluate shopping system search engines. Experiments were also carried out to assure the correctness of the quality index.

Keywords Search engines, Quality factors, Quality index, Quality evaluation, Online shopping search

1 引言

网购零售平台是指一类网络平台,它为用户提供如下服务:通过互联网搜索商品信息,通过电子订单发出购物请求,在商品提供商及购物者达成一致协议后,通过约定的付款方式,以快递或当面交易等方式进行交易。

通常,购物平台会将一小部分可能被广大用户购买的商品显示在购物网站的首页,因此这些商品得到了很大程度的曝光。然而剩下的很大一部分商品并没有显示在首页,需要用户通过关键字搜索,或者分类查看商品等行为才能获取到相关的商品信息。商品搜索功能是网络购物平台提供的最主要的功能之一,用户可以根据自己的需求设置关键字进行搜

索,同时可以根据一些如标签、筛选等类型的操作对搜索的结果进行分类查看,如根据类别、价格、发货地址等因素对商品进行筛选显示,根据商家信用或销量等对商品进行排序,从而满足自己寻找特定商品的需求。

因此,网购平台作为典型的大数据系统,其商品搜索功能的好坏对网购平台的质量有很大的影响。商品搜索功能的好坏不仅会影响用户体验、网购平台的形象,而且会在无形中改变用户在线购买商品的行为。一个存在质量问题的网购平台搜索引擎可能导致用户无法获取到自己所需要的商品,这也影响着网购平台的经济效益。

目前,针对网购平台搜索功能,尚没有人提出评估其质量的因素及具体的质量指标。对于网购平台质量的评估,目前

到稿日期:2016-10-11 返修日期:2016-12-12 本文受国家自然科学基金(61402229, 61502233),江苏省博士后基金(1401043B),南京大学软件新技术国家重点实验室开放式基金(KFKT2015B10),江苏省高校自然科学基金项目(15KJB520030)资助。

陈浩(1993—),男,硕士生,主要研究方向为大数据质量保障、云计算等,E-mail:11510600698@njust.edu.cn;陶传奇(1984—),男,博士,硕士生导师,主要研究方向为大数据质量保障、软件质量评估、软件测试与分析等,E-mail:taochuanqi@njust.edu.cn(通信作者);文万志(1982—),男,博士,副教授,主要研究方向为软件分析与测试、软件错误定位,E-mail:wenzhanzhi@ntu.edu.cn。